

# Universität Leipzig

## Institut für Informatik

in Zusammenarbeit mit dem

**Institut für Angewandte Geodäsie, Außenstelle Leipzig**

## Diplomarbeit

4. April 1997

**Thema: Implementation des ATKIS-Datenmodells in GIS mit relationaler  
Attributverwaltung unter dem Aspekt der Fortführung**

vorgelegt von : Jan Ramsch  
geboren am 10.05.1972 in Lauchhammer  
Betreuer : Prof. Dr. E. Rahm, Universität Leipzig  
Dr. M. Endrullis, Institut für Angewandte Geodäsie Leipzig  
Dr. D. Sosna, Universität Leipzig

*Ein Gorilla und ein Zweig. Damit fing der ganze Ärger an.*

Douglas Adams

## Vorwort

Digitale Geobasisdaten sind in den letzten Jahren zu einer wichtigen Grundfesten unserer Informationsgesellschaft avanciert. Die Anwendungsgebiete sind mannigfaltig: Verwaltung, Planung, Verkehrssteuerung, Navigation und Umweltschutz sind nur einige Beispiele für Bereiche, in denen die Arbeit ohne Verwendung von Geobasisdaten heute nicht mehr denkbar ist. In diesen stark im Wachsen begriffenen Sektor ordnet sich diese Arbeit ein. Sie ist am Institut für Informatik der Universität Leipzig in enger Zusammenarbeit mit dem Institut für Angewandte Geodäsie, Außenstelle Leipzig, entstanden und widmet sich der Verwirklichung und Weiterentwicklung des Amtlichen Topographisch-Kartographischen Informationssystems der Bundesrepublik Deutschland, ATKIS. Sie vereinigt Aspekte der Forschungsbereiche Datenbanken, Kartographie, Geodäsie und Geoinformationssysteme. Einen besonderen Schwerpunkt bilden temporale Datenmodelle.

Es ist nicht Anliegen dieser Arbeit, eine Implementation der ATKIS-Datenbank zu schaffen. Vielmehr wird der Versuch unternommen, die Anforderungen von ATKIS mit klassischen kartographischen Gesichtspunkten und modernen Datenbanktechniken zu vereinen und ein schlüssiges Konzept für ein zukunftsweisendes topographisches Informationssystem vorzulegen. Dabei sollen Aspekte der Fortführung geographischer Datenbestände, der Abbildung historischer Entwicklungen, zeitveränderliche Datenstrukturen auf Basis eines Metainformationssystems und der Modellierung komplexer topographischer Zusammenhänge verwirklicht werden.

An dieser Stelle möchte ich mich bei allen, die zur Entstehung dieser Arbeit einen Beitrag geleistet haben, herzlich bedanken. Besonders danken möchte ich meinen Mentoren Dr. Endrullis vom IfAG und Dr. Sosna von der Uni Leipzig für ihre ausgezeichnete und geduldige Betreuungsarbeit, Prof. Dr. Rahm, Uni Leipzig, für die Übernahme der Schirmherrschaft über diese Arbeit, Dr. Hoppe und Dr. Plötner, IfAG, für ihre hervorragende fachliche und korrekturleserische Unterstützung, Frau Löffler, IfAG, für Programmierhilfe, warmen Tee und tatkräftige Hilfe, Herrn Weise, IfAG, für umfassende Hilfe bei der Bewältigung technischer Probleme, Frau sskdfjhg, IfAG, für moralischen Rückenhalt, den beiden Praktikanten Weber und Ruhstorfer für viel Geduld („Morgen Leute, wir müssen was ändern!“), sowie allen anderen vom IfAG, die nicht unwesentlich zum Gelingen dieser Arbeit beigetragen haben (Nicht unterkriegen lassen!), dem Internet für seine Existenz, Digital für zeitweise funktionierende Rechentechnik, meinen Eltern für finanzielle und moralische Unterstützung („Werde endlich mal fertig!“), und nicht zuletzt Dana für viel Geduld, Plätzchen und Korrekturlesen.

# Inhaltsverzeichnis

Inhaltsverzeichnis	i
<b>1 Einleitung</b>	<b>1</b>
<b>2 ATKIS</b>	<b>3</b>
2.1 Motivation . . . . .	3
2.2 Grundlagen . . . . .	4
2.2.1 Digitales Landschaftsmodell (DLM) . . . . .	5
2.2.2 Digitales Kartographisches Modell (DKM) . . . . .	7
2.3 Kataloge . . . . .	9
2.3.1 Objektartenkatalog . . . . .	9
2.3.2 Signaturenkatalog . . . . .	10
2.4 Datenmodell . . . . .	10
2.5 Abgabe und Fortführung von ATKIS-Daten . . . . .	13
2.6 Praktische Umsetzung . . . . .	14
<b>3 ARC/INFO</b>	<b>17</b>
3.1 INFO . . . . .	17
3.2 ARC und ARCEDIT . . . . .	17
3.2.1 Route . . . . .	18
3.2.2 Region . . . . .	19
3.2.3 Feature Attribut Tables . . . . .	19
3.3 Database Integrator . . . . .	19
3.4 ArcStorm . . . . .	20
<b>4 Temporale Datenmodelle</b>	<b>21</b>
4.1 Wege . . . . .	21
4.2 Entwicklung Temporaler Datenmodelle . . . . .	22
4.3 Temporale Grundbegriffe . . . . .	23
4.3.1 Zeit . . . . .	23
4.3.2 Geschichten . . . . .	25
4.4 Geschichten in ATKIS . . . . .	28
4.4.1 Geschichten spatialer Daten . . . . .	28
4.4.2 Geschichten aspatialer Daten . . . . .	29
4.5 Versionen . . . . .	30
4.6 Versionen im relationalen Modell . . . . .	33
4.6.1 Unterscheidung bitemporal erweiterter Relationenmodelle . . . . .	34

4.7	Temporales ATKIS–Datenmodell . . . . .	36
4.8	Temporale Operationen für ATKIS . . . . .	37
4.8.1	Änderungsoperationen für aktuelle Tupel . . . . .	37
4.8.2	Änderungsoperationen für historische Tupel . . . . .	37
4.9	Zusammenfassung . . . . .	38
4.10	Weiterführende Literatur . . . . .	38
<b>5</b>	<b>Versionen von Metadaten</b>	<b>39</b>
5.1	Allgemeine Begriffe . . . . .	39
5.2	Metadaten von zeitveränderlichen Daten . . . . .	40
5.3	Zusammenfassung . . . . .	41
<b>6</b>	<b>Datenstrukturtabellen</b>	<b>42</b>
6.1	Anforderungen an die Datenstrukturtabellen . . . . .	42
6.1.1	ATKIS–fremde Objektarten . . . . .	43
6.2	Datenmodell . . . . .	46
6.3	Realisierung . . . . .	48
6.3.1	Bezeichnungsfehler (1) . . . . .	49
6.3.2	Zusammenfassungsfehler (2) . . . . .	51
6.3.3	Inhaltliche Fehler (3) . . . . .	53
6.3.4	Unvollständigkeiten (4) . . . . .	53
6.3.5	Sonstige Fehler(5) . . . . .	54
<b>7</b>	<b>DLM–Datenmodell</b>	<b>55</b>
7.1	Spatiale DLM–Daten . . . . .	55
7.1.1	Einfaches Geometriemodell . . . . .	56
7.1.2	Komplexes Geometriemodell . . . . .	57
7.2	Attributive DLM–Daten . . . . .	59
7.3	Objekte und Objektteile . . . . .	61
7.3.1	Referenzen zwischen Objektteilen . . . . .	63
7.4	Komplexobjekte . . . . .	64
7.4.1	Komplexobjektart klassifizierte Straße . . . . .	64
7.4.2	Weitere IfAG–interne Komplexobjektarten . . . . .	66
7.4.3	Bemerkungen zu den IfAG–internen Komplexobjektarten . . . . .	67
7.5	DLM–Datenmodell im Überblick . . . . .	67
7.6	Implementation . . . . .	68
7.7	weitere Etappen . . . . .	69

<b>8 Modellierungsregeln für ATKIS–Objekte</b>	<b>73</b>
8.1 Attribute . . . . .	73
8.2 Netzbildung . . . . .	74
8.3 Objektteilbildungsregeln . . . . .	75
8.3.1 punktförmige Objektteile . . . . .	77
8.3.2 linienhafte Objektteile . . . . .	77
8.3.3 flächenhafte Objektteile . . . . .	77
8.4 Objektbildungsregeln . . . . .	78
8.5 Regeln zu Objektarten . . . . .	78
8.5.1 Regeln zu weiteren Objektarten . . . . .	82
<b>9 Zusammenfassung</b>	<b>83</b>
<b>Tabellenverzeichnis</b>	<b>vi</b>
<b>Abbildungsverzeichnis</b>	<b>vii</b>
<b>Abkürzungsverzeichnis</b>	<b>viii</b>
<b>Glossar</b>	<b>x</b>
<b>Literaturverzeichnis</b>	<b>xvi</b>
<b>Literatur zu temporalen Datenmodellen</b>	<b>xix</b>
<b>Internetadressen zu temporalen Datenmodellen</b>	<b>xxvii</b>
<b>Internetadressen zu Geoinformationssystemen</b>	<b>xxvii</b>
<b>Erklärung</b>	<b>xxviii</b>
<b>A Datenstrukturtabellen</b>	<b>xxix</b>
A.1 IfAG–interne Objektarten . . . . .	xxix
A.2 Attributklassen . . . . .	xxxi
A.3 SQL–Aufrufe zur Generierung der DST . . . . .	xxxix
A.4 Komponenten der Datenstrukturtabellen . . . . .	xli
A.5 Unstimmigkeitenkatalog . . . . .	xliii

## B Tools für die ATKIS-Datenbank

xlvi

B.1 db_report . . . . .	xlvi
B.2 create_atkis_ex . . . . .	xlix
B.3 atkisoj . . . . .	li

ESRI® und ARC/INFO® sind eingetragene Warenzeichen von Environmental Systems Research Institute, Inc.

ARCEDIT, ArcStorm, ARC Macro Language (AML) und DATABASE INTEGRATOR sind Warenzeichen von Environmental Systems Research Institute, Inc.

INFORMIX® ist ein eingetragenes Warenzeichen der ???

# 1 Einleitung

Diese Arbeit setzt sich mit einem komplexen und vielschichtigen Thema auseinander. Dabei werden mehrere auf den ersten Blick unabhängige Forschungsgebiete berührt und im Kontext von ATKIS miteinander vereint. Klassische Datenbankkonzepte und die speziellen Anforderungen an Geoinformationssysteme werden mit kartographischen und topologischen Gesichtspunkten ebenso verbunden wie mit innovativen temporalen Datenmodellen und Metainformationssystemen.

In diesem Sinne sind in Abbildung 1 die primären Elemente des Projektes ATKIS am Institut für Angewandte Geodäsie dargestellt. Entsprechend dieser Aufteilung widmen sich die einzelnen Kapitel dieser Arbeit jeweils einem der angegebenen Gebiete. Diese Abgrenzung kann jedoch nicht vollständig gelingen, da sich die einzelnen Themen überschneiden und gegenseitig bedingen und beeinflussen.

Das Kapitel *ATKIS* gibt eine Einführung in die Themenkomplexe ATKIS, Geomodellierung und Fortführung von Geodaten. Dabei werden bereits grundsätzliche Anforderungen an eine Implementation des DLM-Datenmodells aufgestellt. Im Rahmen der Diskussion der Fortführung und Weitergabe von ATKIS-Daten wird die Notwendigkeit alternativer Datenmodelle erkannt.

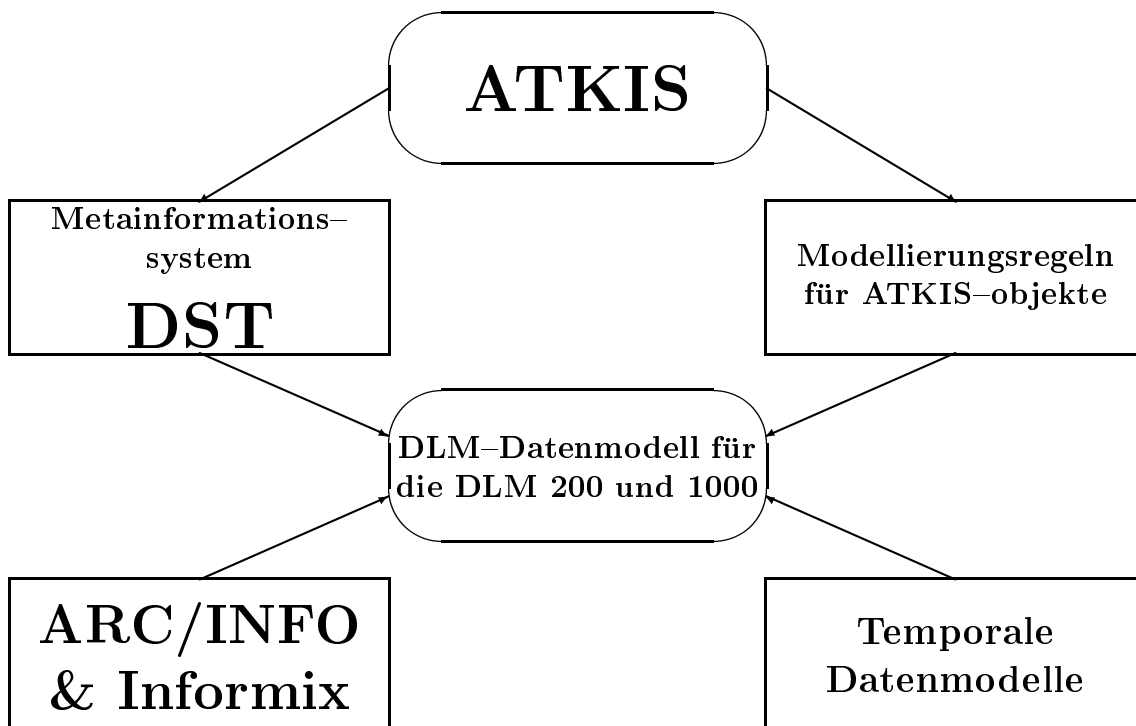


Abbildung 1: Komponenten der Diplomarbeit



Im Kapitel *ARC/INFO* werden die technischen Möglichkeiten für eine Implementation des DLM-Datenmodells unter dem Geoinformationssystem ARC/INFO diskutiert. Dabei wird die grundsätzliche Fähigkeit von ARC/INFO nachgewiesen, die komplexen geometrischen Aspekte des DLM-Datenmodells vollständig und effizient abzubilden. Zusätzlich wird die Bevorzugung eines modernen relationalen Datenbankmanagementsystems für die Verwaltung der Objektstrukturen und attributiven Daten begründet.

Die im ersten Kapitel angesprochenen Probleme bei der Fortführung und Weitergabe von ATKIS-Daten werden im Kapitel *Temporale Datenmodelle* aufgegriffen und ausführlich diskutiert. Als Konsequenz dieser Auseinandersetzung wird anschließend eine Einführung in den Bereich der temporalen Datenmodelle gegeben. Die vorgestellten Konzepte werden bezüglich ihrer Tauglichkeit für die Anwendung im Rahmen einer ATKIS-Implementation untersucht.

Die für die Schaffung einer effizienten ATKIS-Implementation unerlässlichen Struktur- und Hintergrundinformationen über die DLM-Datenstruktur werden in dem speziell für diesen Zweck entwickelten Metainformationssystem *Datenstrukturtabellen* (DST) abgelegt und verwaltet. Vor dem gleichnamigen Kapitel, welches sich mit allen konzeptionellen und praktischen Fragen der DST auseinandersetzt, werden im Kapitel *Versionen von Metadaten* die temporalen Aspekte eines solchen Metainformationssystems betrachtet.

Das eigentliche Anliegen dieser Arbeit, die Schaffung eines leistungsfähigen Datenmodells für die Verwaltung und Fortführung von DLM-Daten, wird in Kapitel *DLM-Datenmodell* dokumentiert. Fragen der Geometriemodellierung und der Verwaltung attributiver Daten unter Vermeidung von Redundanzen kommen hier ebenso zur Sprache wie die Umsetzung komplexer Objektstrukturen und spezielle Implementierungsfragen.

Im Kapitel *Modellierungsregeln für ATKIS-Objekte* werden schließlich exemplarisch am Beispiel des Objektbereiches Siedlung die während dieser Arbeit aufgestellten und zusammengetragenen Modellierungsregeln vorgestellt.

Abschließend soll noch auf die umfangreichen Anhänge verwiesen werden, die neben einer Reihe von für das Verständnis und die Weiterentwicklung dieser Arbeit wichtigen Tabellen und Aufstellungen auch Beschreibungen der im Rahmen dieser Arbeit entstandenen Programme und Tools enthalten.

## 2 ATKIS

Amtliches Topographisch–Kartographisches Informationssystem (ATKIS) – so heißt das Vorhaben der Landesvermessungsverwaltungen zum Aufbau Digitaler Landschaftsmodelle und Digitaler Kartographischer Modelle. Es entstand im Jahre 1985 als Projekt der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV) und hatte eine 1989 herausgegebene Gesamtdokumentation zum Ergebnis, welche die Konzeption, die logischen Strukturen (Datenmodell und Datenaustauschformat) sowie Objektarten- und Signaturenkataloge enthält.

Nach den Vorstellungen der AdV wird ATKIS einmal einen einheitlichen digitalen topographischen Datenbestand über die dreidimensionale Struktur (Topologie) der Erdoberfläche Deutschlands enthalten. Diese immense Informationsmenge soll interessenneutral geführt werden und als staatliche Dienstleistung für öffentliche und private Interessenten zugänglich gemacht werden. Dabei sollen alle Daten authentisch und aktuell gehalten werden. Die vollständige Verwirklichung aller Aspekte dieser Idee liegt zum jetzigen Zeitpunkt aber noch in unabsehbarer Zukunft.

ATKIS besteht hauptsächlich aus zwei Teilen, den Digitalen Landschaftsmodellen (DLM) und den Digitalen Kartographischen Modellen (DKM) (siehe Abb. 2.2.1 und 2.2.2). In jüngerer Zeit werden auch verstärkt Rasterdaten wie die Digitale Topographische Karte (DTK) in ATKIS eingebracht.

Für die praktische Umsetzung von ATKIS existieren bereits detaillierte Vorstellungen. Es wird empfohlen, das ATKIS–Datenmodell auf der Grundlage der ALK–Grundrißdatei systemunabhängig zu realisieren. Bei der ALK handelt es sich um die *Automatisierte Liegenschaftskarte*, welche in Deutschland bereits für große Flächen vorliegt. Weiterhin wird empfohlen, als Datenaustauschformat für ATKIS die einheitliche Datenbankschnittstelle EDBS des Vorhabens ALK zu nutzen. Der Aufbau von ATKIS soll stufenweise erfolgen, wobei besonderes Augenmerk auf das DLM zu richten ist.

### 2.1 Motivation

Bis in die siebziger Jahre existierten Karten über die Topologie der Erdoberfläche fast ausschließlich in analoger Form. Das entsprach zum damaligen Zeitpunkt auch den Anforderungen der Anwender. Die zu dieser Zeit einsetzende und bis heute anhaltende rasante Entwicklung komplexer Informationssysteme und graphischer Datenverarbeitung eröffnete aber nicht nur die Möglichkeit, authentische topographische Informationen in digitaler Qualität bereitzustellen, auch der Bedarf der Anwender an aktuellen und verarbeitbaren Daten stieg an. Die seit den achtziger Jahren im IfAG entwickelten Techniken und Werkzeuge für die Rasterdatenverarbeitung und –fortführung versetzen die Landesvermessungsämter heute

in die Lage, topographische Kartenwerke effizient digital zu erstellen und zu aktualisieren. Dadurch wurde die klassische analoge Kartenbearbeitung für eine Reihe von Kartenwerken bereits abgelöst.

Bereits Mitte der achtziger Jahre wurde parallel dazu von den Landesvermessungsbehörden die Notwendigkeit der Bereitstellung topographischer Vektordaten erkannt. Im Jahr 1985 wurde deshalb das Projekt ATKIS ins Leben gerufen. Die Landesvermessungsbehörden stellten sich damit die Aufgabe, einen Standard für die einheitliche Erfassung eines plattformabhängigen, universell verwendbaren Datenbestandes zu schaffen, welcher die Bedürfnisse der Verwaltung, des Rechtsverkehrs, der Wirtschaft, des Verkehrs, der Landesplanung, der Bauleitplanung und Bodenordnung, der Verteidigung, der Forschung und vieler anderer Bereiche befriedigt. Dies erforderte die Erarbeitung eines bundesweit tragbaren Konzepts, das folgende Komponenten vereinigen mußte:

- Spezifikation der logischen Datenstruktur für die objektbasierte Gliederung, Erfassung, Speicherung und Abbildung der Landschaft in einem digitalen Modell und für die graphische Ausgabe in Karten,
- Erarbeitung einer logischen Datenstruktur für den Austausch digitaler topographischer und kartographischer Informationen,

Es beinhaltet aber *nicht* die Realisierung eines Programmsystems. Die genannten Bestandteile sollen mehrere Bedingungen sicherstellen:

- Möglichkeit einer einfachen und zuverlässigen Beschaffung digitaler topographischer Daten, welche mit Fachdaten verknüpft werden können,
- Authentizität und Aktualität der Daten,
- Wirtschaftlichkeit der Erfassung und Weiterverarbeitung durch Standardisierung.

Im Hinblick auf diese Situation beschlossen die Landesvermessungsämter in Zusammenarbeit mit dem Institut für Angewandte Geodäsie (IfAG) die Bildung einer Arbeitsgruppe, welche in den folgenden Jahren ATKIS entwickelte.

## 2.2 Grundlagen

Die Grundlage von ATKIS bildet die in der modernen Kartographie vertretene Modelltheorie. Danach entsteht durch die topographische Landesaufnahme zunächst ein topographisches Landschaftsmodell auf Grundlage der Landschaft mit ihren topographischen Objekten und deren Eigenschaften. Das dreidimensionale Erscheinungsbild der Erdoberfläche wird strukturiert, d.h. nach topologischen Objekten und Objektteilen und deren Beziehungen

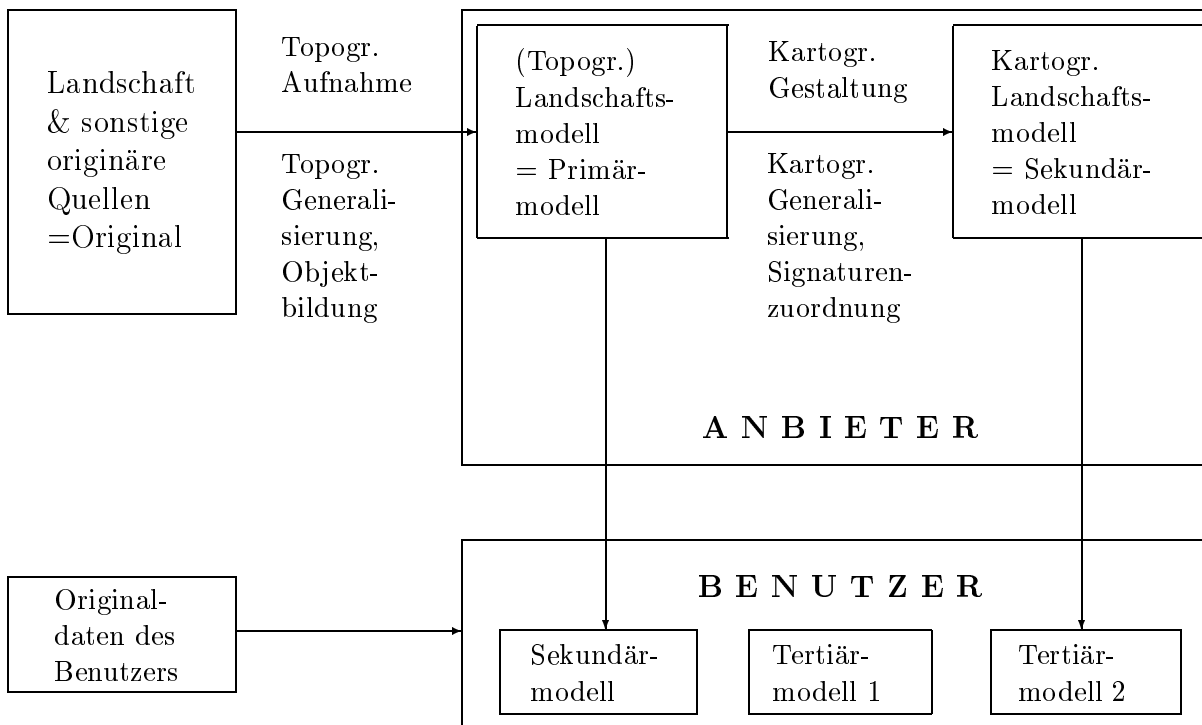


Abbildung 2: Zusammenhang zwischen Primär-, Sekundär-, und Tertiärmodell der Landschaft

gegliedert. Diese Objekte werden nach Form, Lage und topologischen Relationen erfasst, den im Objektartenkatalog (OK) aufgeführten Objektarten zugeordnet, durch Attribute beschrieben, verschlüsselt und gespeichert. Auf diese Weise entsteht das DLM.

Als nächste Stufe werden diese Daten in das Sekundärmodell DKM überführt. Hierbei werden unter Berücksichtigung kartographischer Generalisierungsprozesse und unter Verwendung des Signaturenkataloges (SK) den im DLM gespeicherten topographischen Objekten Kartenobjekte (Signaturen) zugeordnet (siehe Abb. 2 und 3). Aus dem DKM können schließlich unter Anwendung kartographischer Darstellungsregeln Kartenwerke und andere Ausgabeformen gewonnen werden. Die in diesem Ausgabeprozess in der Regel gewonnenen Rasterdaten mit dem bereits vollständigen Kartenbild werden als Digitale Topographische Karten (DTK) bezeichnet.

### 2.2.1 Digitales Landschaftsmodell (DLM)

Das DLM beinhaltet topographische Objekte, welche maßstabsfrei nach Lage und Form mit Hilfe von Landeskoordinaten (Gauß-Krüger-Koordinatensystem) und Landeshöhen (über Normal Null) geometrisch beschrieben, jedoch *nicht* in kartographische Zeichen umgesetzt sind. Diese werden nach sachlogischen Merkmalen in Objektarten gegliedert und zusammengefaßt, welche wiederum folgenden Objektbereichen zugeordnet werden:

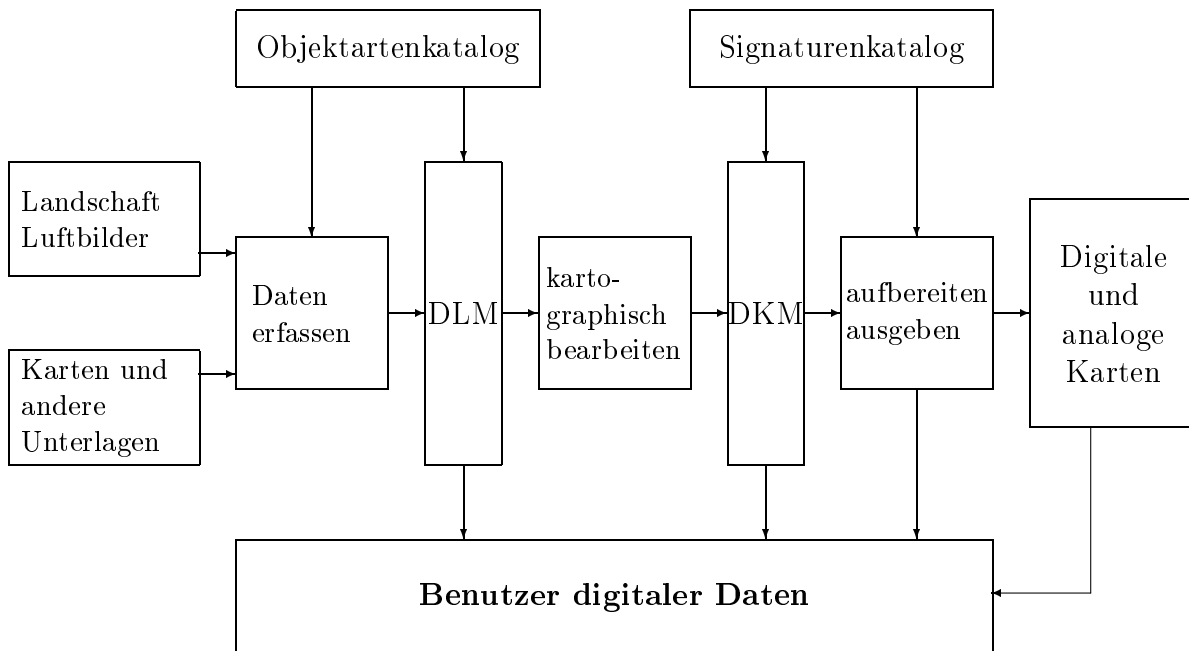


Abbildung 3: Informationsfluß in ATKIS

1. Festpunkte
2. Siedlung
3. Verkehr
4. Vegetation
5. Gewässer
6. Relief
7. Gebiete

Ein Objekt im DLM ist ein geometrisch begrenzter, durch Namen und Attribute beschriebener Teil der Landschaft. Die Attribute beinhalten quantitative und qualitative Eigenschaften des Objektes. Es besteht aus Objektteilen, die punktförmig, linienförmig oder flächenförmig sein können. Dabei kann ein Objekt mehrere Objektteile enthalten.

Theoretisch könnte die Landschaft maßstabsunabhängig in einem einzigen DLM abgebildet und gespeichert werden. Die Gewinnung der DKM ist aber besonders im kleinmaßstäbigen Bereich aufgrund der notwendigen und zum jetzigen Zeitpunkt noch nicht in genügender Weise automatisierten Generalisierung äußerst problematisch. Deshalb entschied man sich für die Schaffung von drei DLM, welche in ihren Eigenschaften in etwa den gegenwärtigen Kartenwerken des großen, mittleren und kleinen Maßstabbereichs entsprechen (siehe auch Abb. 4):

DLM 25: deckt den Bedarf fast aller privaten, öffentlichen und militärischen Bereiche, insbesondere mit einer Lagegenauigkeit von 3m auch die Anforderungen ziviler und militärischer Navigations- und Verkehrsleitsysteme

DLM 200: vor allem für öffentliche Verwaltungen, insbesondere für Raum- und Verkehrsplanung sowie militärische Belange

DLM 1000: entspricht Spezifizierung der Entwicklung der Digital Cartographic Data Base of Europe 1:1 Mio (EURODB)

### 2.2.2 Digitales Kartographisches Modell (DKM)

Die ATKIS-Begründer charakterisieren das DKM wie folgt:

Ein Digitales Kartographisches Modell enthält in digitalisierter Form die Gesamtheit der in ein zweidimensionales kartographisches Zeichensystem umgesetzten und gespeicherten topographischen Informationen. Es besitzt im Gegensatz zum DLM einen Maßstab und ermöglicht die Ausgabe als analoge Karte. [ATKIS]

Einen etwas besseren Eindruck über die Bedeutung des DKM vermittelt Harbeck in [Har93]:

Digitale Kartographische Modelle beschreiben die Landschaft mit Hilfe digitaler kartographischer Daten. Diese bestimmen nach Anwendung kartographischer Generalisierungsregeln für die topographischen Objekte Positionen, Achsen und Flächen, denen eine kartographische Signatur zugeordnet ist. Die geometrischen Dimensionen dieser Kartenobjekte beziehen sich auf einen Kartenmaßstab, auf ein Abbildungssystem und auf das Koordinatensystem der Landesvermessung. Ihre Identitäts- und Signaturenmerkmale sind alphanumerisch kodiert. Welche Kartenobjekte ein DKM ausmachen und wie sie zu signaturieren sind, ist im ATKIS-Signaturenkatalog festgelegt.

Die weitverbreitete Ansicht, daß ein DKM bereits ein herkömmliches Kartenwerk darstellt, erweist sich bei näherer Betrachtung als falsch. Tatsächlich ist die Karte nur eine von mehreren Ausgabemöglichkeiten, wenn auch die wichtigste. Das DKM selbst besitzt eine komplexe Struktur ähnlich der des DLM mit Objekten, Objektteilen und Komplexobjekten.

Ein DKM wird aus dem DLM abgeleitet, indem nach der Anwendung gewisser Generalisierungsregeln und anderer Aufbereitungsschritte, während derer auch Objekte zusammengefaßt oder gestrichen werden können, jedem verbliebenem Objekt des DLM ein

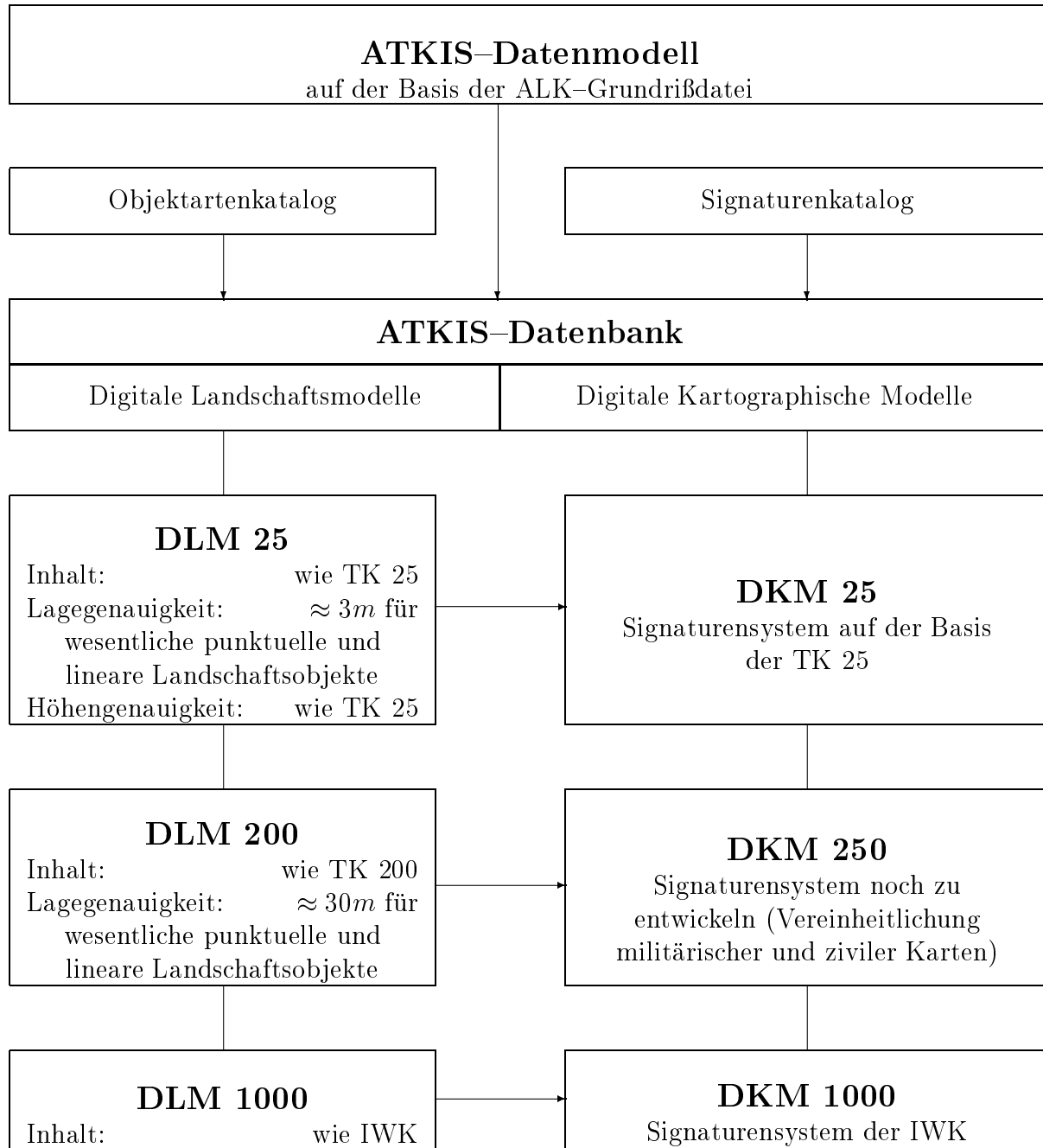


Abbildung 4: Die ATKIS-Komponenten und ihre Beziehungen [ATKIS]

DKM-Objekt zugeordnet wird. Dabei kommen die im Signaturenkatalog (siehe Abschnitt 2.3.2) festgelegten Regeln zur Anwendung.

Da das DKM<sup>1</sup> nicht im Aufgabengebiet dieser Arbeit liegt, sollte diese kurze Einführung ausreichend sein. Nähere Informationen sind in der ATKIS-Dokumentation [ATKIS] und den weiterführenden Empfehlungen der Arbeitsgruppe Kartographie (AA Kart) der AdV (zum Beispiel [AKK93]) enthalten.

## 2.3 Kataloge

### 2.3.1 Objektartenkatalog

Der Objektartenkatalog enthält detaillierte Vorschriften für die Erfassung, attributive und namentliche Beschreibung, Kodierung und Objektbildung von topographischen Objekten der Landschaft und legt damit den Inhalt des DLM fest. Für jedes DLM existiert ein OK. Diese werden mit OK 25, OK 200 und OK 1000 bezeichnet.

Jeder OK ist nach Objektarten gegliedert, welche zu Objektgruppen zusammengefaßt werden, die wiederum Objektbereichen angehören. Zu jeder Objektart enthält der OK eine Liste der zu erfassenden Attribute, der erlaubten Attributwerte einschließlich eventueller alphanumerischer Kodierungen, eine Liste der erlaubten Referenzen, Modelltyp, Erfassungskriterien und die exakte Definition der Objektart. Außerdem können spezielle Modellierungsregeln und Objektteil- und Objektbildungsregeln aufgeführt sein. Solche Regeln sind auch für Objektgruppen oder ganze Objektbereiche definiert.

Die Attribute der Objektarten können drei verschiedenen Typen angehören. Die am häufigsten anzutreffende Art sind die *kodierten Attribute*. Bei Ihnen werden für das DLM letztendlich nicht die eigentlichen Attributwerte, sondern ihre im OK angegebenen vierstelligen alphanumerischen Kodierungen gespeichert. Diese sind auch die einzigen erlaubten Werte.

Manche Attribute wie geographische Namen und Maßangaben können beliebige Werte enthalten. Man spricht dabei von *stetigen Attributen*. Eine etwas irreführende Begriffsbildung, die sich aber im Sprachgebrauch eingebürgert hat und deshalb beibehalten werden soll.

Die *klassifizierten Attribute* schließlich enthalten Werte, die für einen bestimmten Bereich von numerischen Werten stehen. Es findet sozusagen eine Abbildung des Zahlenbereiches auf einen Attributwert statt.

Die OK existieren nicht etwa autonom, sie sind im Gegenteil sehr eng miteinander verbunden. So fungiert der OK 25 als Grundlage für den OK 200, und dieser wiederum als

---

<sup>1</sup> Es ist eigentlich falsch, von *dem* DKM zu reden, da es beliebig viele DKM geben kann. In der relevanten Literatur hat es sich aber eingebürgert, von dem DKM stellvertretend für alle DKM zu sprechen.



Grundlage des OK 1000. Der OK 200 wird beispielsweise aus dem OK 25 abgeleitet, indem alle für den Maßstabsbereich 1 : 200000 irrelevanten Objektarten, Attribute und Attributwerte gestrichen, Attributwerte zusammengefaßt und Modellierungsregeln angepaßt werden. Diese als *Selektionsprinzip* bezeichnete Grundregel ermöglicht theoretisch die vollautomatische Ableitung der DLM 200 und 1000 vom DLM 25. Hier ergeben sich zwei Probleme. Zum einen sind die automatischen Verfahren zur kartographischen Generalisierung noch nicht hinreichend ausgereift, zum anderen ist das Selektionsprinzip entgegen der Konzeption von ATKIS noch nicht konsequent durchgesetzt (siehe Abschnitte 6.3.1, 6.3.2 und 6.3.4). Die Ursache dafür liegt im Bearbeitungsstand der OK 200 und 1000. Es handelt sich hier nur um Arbeitsversionen, die in nächster Zeit von der zuständigen Expertengruppe der AdV überarbeitet und an den OK 25 angepaßt werden. Dabei werden auch die im Rahmen dieser Arbeit entdeckten Unstimmigkeiten in den OK (siehe Anhang A.5) Beachtung finden.

Einen Eindruck vom Aufbau der OK gibt Abbildung 5.

### 2.3.2 Signaturenkatalog

Die Signaturenkataloge (SK) haben die Aufgabe, Landschaftsobjektarten in Kartenobjektarten umzubilden, die Auswirkungen der kartographischen Generalisierung auf die Kartenobjekte zu beschreiben und den Kartenobjektarten Signaturen zuzuordnen. [ATKIS] Sie sind somit Grundlage für die Ableitung der DKM aus den DLM. Die SK sind ähnlich wie die OK in Objektbereiche, Objektgruppen und Objektarten gegliedert und enthalten allgemeine und objektartspezifische Vorschriften für die Ableitung von DKM-Objekte aus DLM-Objekten und die genaue Beschreibung der Kartenobjektarten zugeordneten Signaturen. Die SK sind in konzeptionell mit den herkömmlichen Musterblättern vergleichbar. Eine vollständige Beschreibung sowie die SK selbst sind in [ATKIS] zu finden.

## 2.4 Datenmodell

Das DLM-Datenmodell besteht, wie aus Abb. 6 ersichtlich, aus drei Ebenen, der semantischen, der semantisch-topologischen und der geometrischen Ebene.

Die semantische Ebene (auch Objektebene) enthält die folgenden Komponenten zur Beschreibung der im DLM enthaltenen Objekte:

- Identifikationsnummer
- fachspezifische Namen
- Modelltyp
- Datum (der Erstellung und der letzten Veränderung)
- Attribute laut OK

ATKIS-Objektartenkatalog (ATKIS-OK) Teil D2: ATKIS-OK 200		Seite Blatt 31.2 1 (2)	Stand 01.11.1995
Nr. Objektbereich 3000 VERKEHR	Nr. Objektgruppe 3100 Straßenverkehr		
Nr. Objektart 3102 Weg		Nr. 3102	
Allgemeine Angaben zur Objektart			
<p><b>Definition:</b> Befestigter oder unbefestigter Geländestreifen, der zum Befahren und/oder Begehen vorgesehen ist. Zum Weg gehören auch Seitenstreifen und Gräben zur Wegeentwässerung einschließlich der auf Über- und in Unterführungen verlaufenden Abschnitte.</p> <p><b>Erfassungskriterium:</b> Wege werden nur erfaßt, wenn sie zu topographisch bedeutsamen Zielen führen. Fußfaden in Sumpfgeländen, Wattenwege, Steige und Pfadspuren im Hochgebirge und ähnliche, nicht deutlich sichtbare und nicht gesicherte Wege werden nur dann erfaßt, wenn keine andere direkte Wegeverbindung besteht.</p> <p><b>Objekttyp:</b> linienförmig</p> <p><b>Besondere Objekt- und Objektteilbildungsregeln:</b> Ein neues Objekt ist zu bilden,</p> <ol style="list-style-type: none"> <li>1. wenn sich der Wert des Attributes FKT ändert (herausgehobenes Attribut im Sinne der Objektbildung)</li> <li>2. an einem Netzknoten mit dem Straßennetz</li> <li>3. an einem Netzknoten des Wegenetzes in Bezug auf abgehende Wege (d.h. drei oder mehr abgehende Wege können nicht zu ein und demselben Objekt gehören)</li> </ol>			
Name			
GN	Geographischer Name		
ZN	Zweitname		
KN	Kurzbezeichnung		
Attribute der Kategorie 1			
BEF	Befestigung		
1000	befestigt		
9997	Attribut trifft nicht zu		
FKT	Funktion		
1701	Hauptwirtschaftsweg, Verbindungsweg (Fahrweg)		
1702	Wirtschaftsweg (Feld-, Waldweg)		
1703	Fußweg, Karren- und Ziehweg, Wattenweg		
1709	(Kletter-)Steig im Gebirge		
ZUS	Zustand		
1100	in Betrieb		
1200	außer Betrieb, stillgelegt		
1300	im Bau		
Referenzen			
Objektteil oben:			
'Brücke, Überführung, Unterführung', 'Tunnel'			
Objektteil unten:			
'Brücke, Überführung, Unterführung', 'Tunnel', 'Strom, Fluß, Bach', 'Graben, Kanal (Wasserwirtschaft)', 'Talsperre, Wehr'			
Anmerkung:			
Überführungsreferenzen können unmittelbar auf Verkehrswege oder Gewässer bezogen werden, wenn kein Objekt der Objektart 'Brücke, Überführung, Unterführung' oder 'Tunnel' zwischengeschaltet wird.			

Abbildung 5: Auszug aus dem OK 200

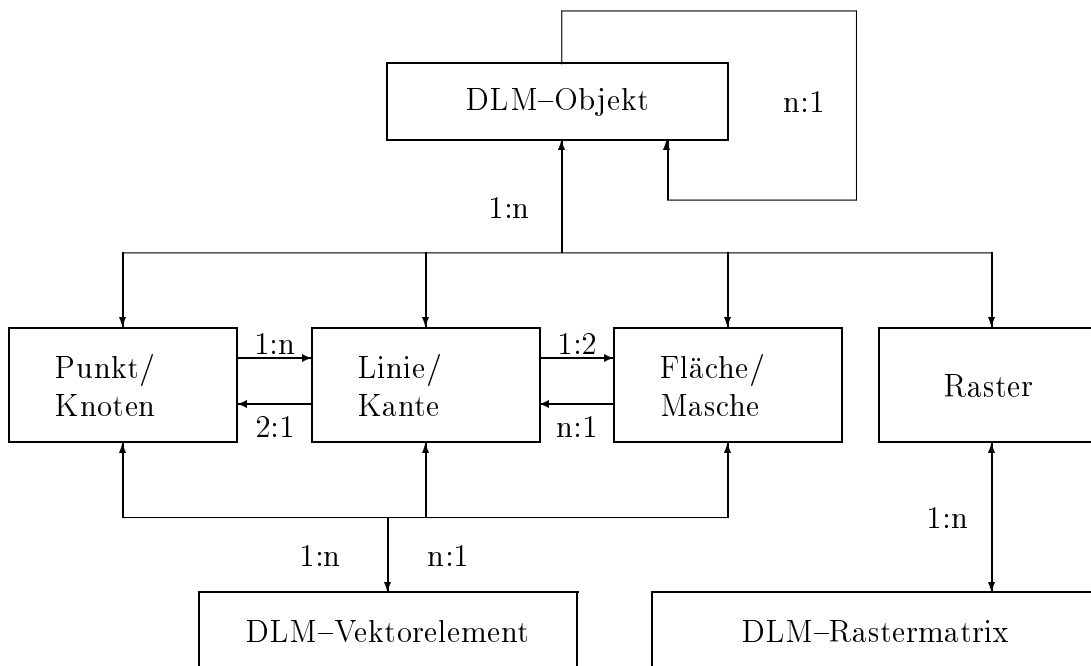


Abbildung 6: Struktur des DLM-Datenmodells

- zusätzliche Attribute (benutzerdefiniert)

Die semantisch-topologische Ebene (in Zukunft Objektteilenebene genannt) enthält alle Objektteile, welche nach semantischen und topologischen Kriterien gebildet werden und vom Typ Fläche/Masche, Linie/Kante, Punkt oder Raster sein können. Sie besitzen analog zu den Objekten eine Identifikationsnummer und Attribute, aber keine Namen und sind durch Referenzen genau einem Objekt und im allgemeinen mehreren Geometrieelementen zugeordnet.

Die im OK aufgeführten Attribute werden der Objektteil- oder der Objektebene zugeordnet. Die Unterscheidung dieser im folgenden Objektteil- bzw. Objektattribute<sup>2</sup> genannten Attribute wird noch von entscheidender Bedeutung sein (siehe Abschnitt 7.2).

Sowohl Objekte als auch Objektteile besitzen, bis auf die Objektkoordinate, noch keinerlei geometrische Informationen. Diese sind erst in der dritten, der geometrischen Ebene, enthalten, welche die geometrische Form der Objektteile, ihre Lage und ihre geometrische Verknüpfung mit anderen Objektteilen beschreibt. Dabei werden alle Geometrieelemente redundanzfrei abgelegt, was durch die Zuordnung mehrerer Objektteile zu einem Geometrieelement ermöglicht wird. Diese Forderung ist besonders für die Fortführung von Datenbeständen wichtig. In den meisten Fällen besteht auch eine semantische Verbindung dieser Objektteile, die auch bei einer Veränderung der Geometrie nicht verlorengehen soll. Typische

<sup>2</sup> Die Objektattribute werden im OK als „hervorgehobene Attribute im Sinne der Objektbildung“ ausgewiesen.

Beispiele sind Grenzen, die auf Verkehrswegen oder Gewässern verlaufen, und verschiedene baulich geprägte Flächen, die mit Industriegebieten zusammenfallen.

Außerdem besteht die Forderung, daß kein Geometrieelement ohne Zugehörigkeit zu einem Objektteil existiert.

Der Aufbau des Datenmodelles des DKM (siehe Abb. 7) scheint zunächst ähnlich dem des DLM zu sein. Es existieren jedoch zwei wesentliche Unterschiede. Der erste besteht in der Existenz von Geometrieelementen zur Objektausgestaltung wie zum Beispiel Texte und Symbole, welche nicht zur geometrischen Lagebestimmung, sondern zur kartographischen Kenntlichmachung dienen. Zweitens existieren im DKM keine attributiven Informationen. Diese werden bei der Überführung des DLM in ein DKM in Signaturen und Texte umgesetzt.

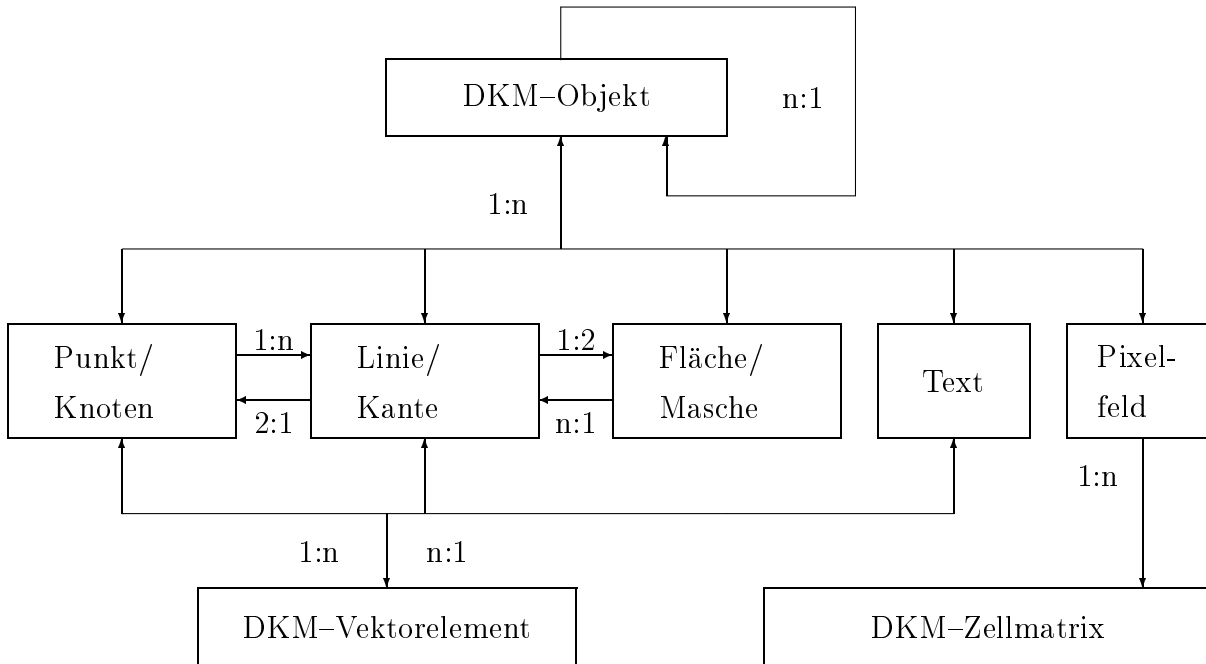


Abbildung 7: Struktur des DKM-Datenmodells

## 2.5 Abgabe und Fortführung von ATKIS-Daten

Der Ablauf von Fortführung und Abgabe von ATKIS-Daten ist schematisch in Abbildung 8 dargestellt. Die Abgabe von ATKIS-Daten (DLM- oder DKM-Daten) an Dritte erfolgt über die *Einheitliche Datenbankschnittstelle* (EDBS) durch das Verfahren *Bezieher Sekundärnachweis* (BZSN). Hier soll nur die Abgabe von DLM-Daten diskutiert werden. Laut [AGH96] ist die angestrebte Verfahrensweise die einmalige Abgabe des gesamten Primärdatenbestandes oder benutzerspezifischer Auszüge (Erstausstattung). Danach werden nur noch die durch Fortführung veränderten, neu erzeugten oder gelöschten Datensätze

abgegeben (*Inkrementelle Fortführungsinformation*). ATKIS-Nutzer müssen durch dieses Verfahren ihre Fachdaten nicht jedesmal neu mit den ATKIS-Daten verknüpfen, es ist lediglich eine Bearbeitung der neu hinzugekommenen und der gelöschten Datensätze notwendig.

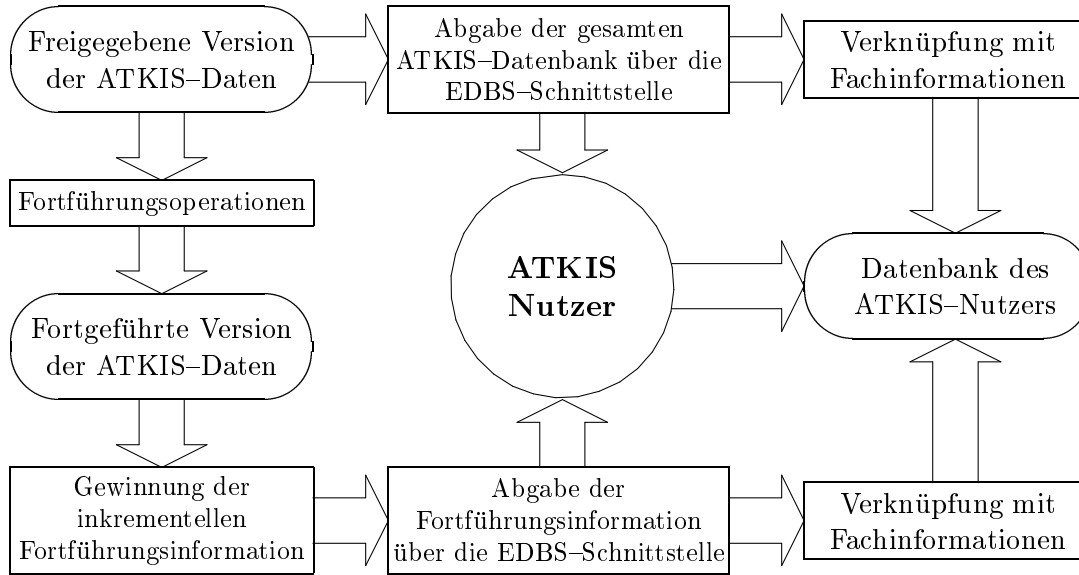


Abbildung 8: Abgabe und Fortführung der ATKIS-Daten

Die Gewinnung der inkrementellen Fortführungsinformation setzt die Verwaltung aller Manipulationen an ATKIS-Objekten- und Objektteilen und eine benutzerspezifische Kontrolle der im Rahmen des BZSN abgegebenen Daten voraus. Die ATKIS-Datenbank ist also um einen Mechanismus zur Verwaltung aller Änderungsoperationen oder adäquater Informationen zu erweitern. Hierfür sind eine Reihe von Verfahren denkbar. Laut [AGH96] werden zur Zeit die Fortführungsdaten für den BZSN gewonnen, indem alle Veränderungen am Primärdatenbestand beziehungsweise gespeichert werden. In Abschnitt 4 werden neben dieser verschiedene andere Möglichkeiten diskutiert. Dabei wird nachgewiesen, daß das dort beschriebene *temporale* ATKIS-Datenmodell den genannten Anforderungen am besten gerecht wird.

## 2.6 Praktische Umsetzung

Bisher wurde das ATKIS-Datenmodell für das DLM 25 auf mehreren Plattformen und unter verschiedenen Systemen realisiert, wobei die Richtlinien der ATKIS-Kommission mehr oder weniger eingehalten wurden. Die folgenden Systeme sind in der Bundesrepublik bereits zum Einsatz gekommen:

**ALK–GIAP der Landesvermessungsverwaltungen**, eingesetzt in Brandenburg, Nordrhein–Westfalen und Schleswig–Holstein<sup>3</sup> unter Nutzung der ATKIS–DB<sup>4</sup>

**Intergraph**, eingesetzt in Bayern, Hessen und Thüringen

**SICAD von Siemens**, eingesetzt in Baden–Württemberg, Mecklenburg–Vorpommern, Niedersachsen, Saarland, Rheinland–Pfalz, Sachsen und Sachsen–Anhalt unter Nutzung der ATKIS–DB oder SICAD–GDB<sup>5</sup>

Die meisten Implementationen verzichten bisher auf eine Abbildung der Höhenkomponente.

Für die Erfassung der topologischen Informationen können verschiedene Informationsquellen genutzt werden, so zum Beispiel Luftbilder, Satellitenbilder, analoges Kartenmaterial wie Katasterkarten und topographische Landeskarten, sowie natürlich die Landschaft selbst. Die Erfassung erfolgt entweder durch vektorielle Digitalisierung, photogrammetrische Auswertung von Luftbildern, Scannen analoger Unterlagen und anschließende Vektorisierung und Objektartenzuordnung oder durch Übernahme und Konvertierung bzw. Aufbereitung bereits vorhandener Datenbestände aus anderen Informationssystemen.

Es haben sich jedoch einige hauptsächliche Informationsquellen herauskristalisiert:

- die Deutsche Grundkarte 1 : 5000 (DGK 5), die Topographischen Karten 1 : 10000 (TK 10) und Orthophotos für das DLM 25,
- die Topographische Übersichtskarte 1 : 200000 (TÜK 200) für das DLM 200<sup>6</sup> und
- die Übersichtskarte 1 : 500000 (ÜK 500) für das DLM 1000.

Die flächendeckende Erfassung des DLM 25 der ersten Aufbaustufe von ATKIS, auch DLM 25/1 genannt, ist für die meisten Bundesländer fast abgeschlossen. Das DLM 25/1 ist das DLM 25 unter Vernachlässigung der Höheninformation und unter Aussparung einiger Objektarten und Attribute. Der Aufbau der DLM 200 und 1000 befindet sich noch im Anfangsstadium.

Im Gegensatz zum DLM 25, das in Verantwortung der jeweiligen Landesvermessungsverwaltungen für jedes Bundesland separat entsteht, obliegt der Aufbau der DLM 200 und

---

<sup>3</sup> Die Produktionswege der einzelnen Länder wurden [Chr92] entnommen.

<sup>4</sup> Mit ATKIS–DB werden hier selbstentwickelte Datenbanken bezeichnet, im allgemeinen unter Nutzung eines herkömmlichen relationalen Datenbanksystems

<sup>5</sup> SICAD–GDB ist die Graphische Datenbank der Firma Siemens–Nixdorf für das CAD–System SICAD.

<sup>6</sup> Die Verwendung analoger Karten für den Aufbau von ATKIS ist nicht unproblematisch, da beispielsweise die Lage von Landschaftsobjekten durch die Anwendung von Generalisierungs– und Symbolisierungsverfahren auf der Karte fehlerhaft dargestellt wird. Die geplante Verwendung des DLM 25 als Grundlage der DLM 200 und 1000 wird diese Probleme beseitigen.

1000 für das gesamte Bundesgebiet allein dem IfAG, eine Aufgabe, in die sich auch diese Arbeit einordnet. Für die Verwaltung der geometrischen ATKIS-Daten wird am IfAG das relationale Geo-Informationssystem ARC/INFO der Firma ESRI Inc. verwendet werden. Die Grundlagen dieses Programmpaketes und alle für die Implementation des DLM-Datenmodells wichtigen Module werden im folgenden Abschnitt kurz vorgestellt.

Die Implementierungsphase des ATKIS-Datenmodells für die DLM 200 und 1000 ist noch nicht abgeschlossen. Einen wichtigen Beitrag zur Vollendung dieser Aufgabe soll diese Arbeit leisten.

## 3 ARC/INFO

Dieser Abschnitt beschreibt die Grundlagen des für die Realisierung der DLM 200 und DLM 1000 verwendeten modularen Geo-Informationssystems ARC/INFO von ESRI Inc..

Zuerst werden die Module ARC, ARCEDIT und INFO vorgestellt. Diese Komponenten dienen der Verwaltung der spatialen und aspatialen Daten und stellen eine breite Funktionalität zur Datenerfassung, -verwaltung und Auswertung zur Verfügung. Des weiteren soll der Database Integrator untersucht werden, er ermöglicht den komfortablen und sicheren Zugang zu einem relationalen DBMS. Schließlich wird noch der Modul ArcStorm betrachtet, ein Tool zur blattschnittfreien Verwaltung spatialer und aspatialer Daten in und außerhalb von ARC/INFO mit Möglichkeiten des Mehrbenutzerbetriebes.

### 3.1 INFO

Die Weiterentwicklung des in ARC/INFO enthaltenen relationalen Datenbanksystems INFO wurde angesichts besserer Konkurrenzprodukte gegen Ende der achtziger Jahre eingestellt. INFO ist aber nach wie vor die primäre Datenbasis von ARC/INFO. Das heißt, daß bis auf direkte Geometriedaten wie Koordinaten alle Daten von INFO verwaltet werden. INFO stellt entsprechend dem damaligen Entwicklungsstand nur einfache Operationen über Tabellen zur Verfügung. Moderne Transaktionskonzepte und Integritätssicherungen sind kaum verwirklicht. Durch die enge Verknüpfung mit ARC und in Verbindung mit den Modulen Database Integrator, Librarian und ArcStorm bietet es aber eine ausreichende Funktionalität für die Haltung von Geometriedaten. Alle anderen Daten sollten mit Hilfe eines modernen DBMS (Database Management System) verwaltet werden.

### 3.2 ARC und ARCEDIT

Für die Verwaltung von Daten stellt ARC/INFO sogenannte Cover<sup>1</sup> zur Verfügung, ein Organisationsmechanismus, welcher nahezu beliebige Geometrietypen und damit verknüpfte Informationen und Daten beinhalten kann. Die einzige Einschränkung besteht darin, daß die Geometrietypen point und poly nicht in einem gemeinsamen Cover verwaltet werden können.

Alle Geometrietypen, also alle Datentypen mit direktem räumlichen Bezug setzen die Abbildung einer oder mehrerer Koordinaten voraus. Unter ARC/INFO bauen alle Geometrietypen auf zwei Basiselementen auf, dem *node* und dem *vertex*, den einzigen Datentypen mit Koordinaten. Es handelt sich hierbei um zwei verschiedene Arten von Punkten, beschrieben durch jeweils eine *x*- und eine *y*-Koordinate. Daraus folgt, daß ARC/INFO nur in der

---

<sup>1</sup> Dies ist der von der originalen Bezeichnung coverage abgeleitete, eingedeutsche Begriff.



Lage ist, zweidimensionale Geo-Daten zu speichern, die Abbildung der dritten Dimension kann nur indirekt über Attribute erfolgen.

Der node bildet die Grundlage des Geometrietypes (der Featureklasse) *point* und ist zusammen mit dem vertex Basiselement des Types *arc*. Ein arc ist ein gerichteter Linienzug, der aus zwei nodes<sup>2</sup> und einer endlichen Menge von vertices besteht, die den Verlauf des Linienzuges definieren. Aus arcs schließlich wird die Featureklasse *poly* (Polygon) aufgebaut, Ein poly ist eine durch einen oder mehrere arc begrenzte, geschlossene Fläche. Sie wird referenziert über einen *label point*, einer speziellen Art des Geometrietyps *point*, der im Inneren der Fläche liegt. Ein poly kann in seinem Inneren Inseln enthalten. Existieren in einem Cover polys, so wird für jeden arc das links und das rechts liegende poly gespeichert. Das führt uns zu einer der wichtigsten Einschränkungen für polys. Ein poly kann nie durch einen arc geschnitten werden, in diesem Fall würde das poly zu zwei neuen polys zerfallen<sup>3</sup>. Ein in das poly lediglich hineinragender arc (dangle) ist jedoch erlaubt.

Wird in einem Cover die Polygontopologie gebildet, so werden alle arcs an gemeinsamen Schnittpunkten aufgetrennt. Dieses und die Tatsache, daß polys nur jeweils die kleinste von arcs begrenzte Fläche besitzen können, führte zur Entwicklung komplexer Geometrietypen, die aus mehreren arcs bzw. polys bestehen können, den *Routen* und *Regionen*.

### 3.2.1 Route

Die *Route* ist ein komplexer, linienförmiger Geometrietyp. Er besteht aus *Sectionen*. Eine Section ist ein Teil eines arc, gekennzeichnet durch eine Länge im verwendeten Maßsystem und einen Anfangspunkt. Dabei kann ein arc Grundlage für mehrere Sectionen sein. Eine Section ist genau einer Route zugeordnet, die jedoch aus mehreren Sectionen bestehen kann. Dabei gibt es keine Einschränkungen für die Lage der den Sectionen zugrundeliegenden arcs, eine Route muß also kein zusammenhängender Linienzug sein. Es müssen sich aber alle an der Route beteiligten Sectionen und arcs im selben Cover befinden.

Die Verwaltung von Routen erfolgt durch benutzerdefinierte Routenklassen. Das sind abgeleitete Klassen des Geometrietypes *Route*, jeweils mit eigener FAT (siehe Abschnitt 3.2.3). Jede Route gehört genau einer Klasse an.

Der Zusammenhang von Routen, Sectionen, arcs und allen anderen für uns wichtigen Geometrietypen ist in Abbildung 9 dargestellt. Der Geometrietyp *point* ist absichtlich nicht in dieser Darstellung enthalten, da er nicht gemeinsam mit den flächenförmigen Typen in einen Cover verwaltet werden kann und daher immer separat behandelt werden muß.

---

<sup>2</sup> Diese beiden nodes bilden den Anfangs- und den Endpunkt des arc (from- und to-node) und geben damit auch die Richtung des arc an.

<sup>3</sup> In der Praxis werden durch Einfügen eines ein poly zerschneidenden arc nicht sofort zwei neue polys gebildet, es wird lediglich die Polygontopologie zerstört. Erst nach Neubilden der Topologie durch einen der Befehle *build* oder *clean* entstehen die neuen polys.

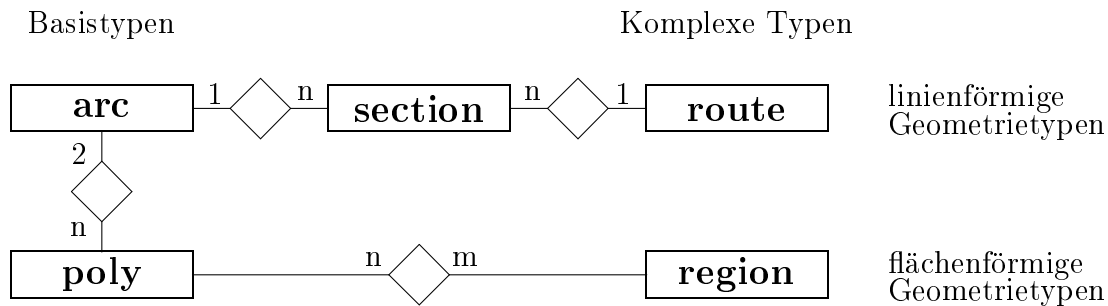


Abbildung 9: Die wichtigsten Geometriotypen in ARC/INFO und ihre Beziehungen

### 3.2.2 Region

Die *Region* ist ein flächenförmiger, komplexer Geometriotyp, der aus polys aufgebaut ist. Dabei besteht eine  $n : m$ -Beziehung, das heißt ein poly kann beliebig vielen Regionen angehören. Es bestehen keine Einschränkungen für die Lage der an einer Region beteiligten polys, sie müssen nicht zusammenhängend sein. Wie bei den Routen sind auch hier zuerst Unterklassen des allgemeinen Geometrietypes Region zu bilden, in denen dann die Regionen verwaltet werden.

### 3.2.3 Feature Attribute Tables

Alle bisher beschriebenen Geometriotypen besitzen neben ihren geometrischen Daten eine sogenannte Feature Attribute Table (FAT). Das sind INFO-Tabellen, in denen zusätzlich zu den standartmäßig enthaltenen Spalten vom Benutzer beliebige Spalten angefügt werden können. Für jedes Geometrieelement existiert eine Zeile in der entsprechenden FAT.

Für jeden in einem Cover enthaltenen Geometriotyp existiert genau eine FAT. Die Namen der FAT (siehe Tabelle 1) sind vom Geometriotyp, dem Covernamen und bei Routen und Regionen vom Namen der Routen- bzw. Regionklasse abhängig.

## 3.3 Database Integrator

Der Database Integrator ermöglicht die komfortable Anbindung eines relationalen DBMS an ARC/INFO. Er stellt Funktionen für die Verknüpfung von INFO-Tabellen und externen RDBMS-Tabellen bereit, unterstützt das Absetzen von SQL-Statements und stellt einen Cursormechanismus zur Verwaltung und Bearbeitung der Ergebnisse von SQL-Queries zur Verfügung.

Der Database Integrator unterstützt die Anbindung eines RDBMS mit Client-Server-Technologie und stellt somit ein zeitgemäßes und ausreichendes Hilfsmittel für die Erweite-

Geometrietyp	FAT-Name	Beschreibung
arc	<cover>.AAT	Arc Attribute Table
node	<cover>.NAT	Node Attribute Tabel
point	<cover>.PAT	Point Attribute Table
poly	<cover>.PAT	Polygon Attribute Table
region	<cover>.PAT<Regionenklasse>	Region Attribute Table
route	<cover>.RAT<Routenklasse>	Route Attribute Table
section	<cover>.SEC<Routenklasse>	Section Attribute Table

Tabelle 1: Die Namen der Feature Attribute Tables [ARC]

rung einer INFO-Datenbank um externe Tabellen dar.

### 3.4 ArcStorm

Der Modul ArcStorm des ARC/INFO-Paketes wurde im Rahmen dieser Arbeit nicht näher untersucht. Er soll hier aber Erwähnung finden, da er als Tool für die blattschnittfreie Verwaltung sehr großer geometrischer Datenmengen für den Aufbau der DLM 200 und 1000 große Bedeutung erlangen könnte. Für die Phase der Implementierung und Testung des DLM-Datenmodells mit kleinen Datenmengen erwies sich ArcStorm aufgrund seiner Forderung nach einer festen, unveränderbaren Datenstruktur als ungünstig.

In Abschnitt 2.5 wurde bereits über die Notwendigkeit der Aufzeichnung von Veränderungen am DLM-Datenbestand gesprochen. ArcStorm bietet prinzipiell eine Möglichkeit der Aufzeichnung älterer Datenbankzustände. Das hier verwendete Verfahren basiert auf der Duplizierung der gesamten Datenbank nach jeder Änderungsoperation. Diese Vorgehensweise ist nicht optimal, besonders bei großen Datenmengen ergeben sich hier enorme Nachteile. Der nachfolgende Abschnitt setzt sich mit dieser Problematik intensiv auseinander und zeigt einen leistungsfähigeren Lösungsansatz auf.

## 4 Temporale Datenmodelle

In Abschnitt 2.5 wurde die Notwendigkeit nachgewiesen, für die im Rahmen des BZSN bereitzustellende Inkrementelle Fortführungsinformation die ATKIS-Datenbank um Informationen über neu erzeugte, veränderte oder gelöschte Datensätze zu erweitern. Neben dieser Forderung existieren von Seiten des IfAG weitere Anforderungen an ATKIS, die so nicht in [ATKIS] festgehalten sind, wie zum Beispiel die Rekonstruktion des ATKIS-Datenbank-Zustandes zu einem früheren Zeitpunkt und die Unterscheidung von korrekten und erfaßten Daten bezüglich eines bestimmten Zeitpunktes. Nach einer Diskussion prinzipieller Lösungswege werden in den folgenden Abschnitten verschiedene Ansätze, Modelle und Lösungsvorschläge für temporale Datenmodelle und Versionenverwaltungen vorgestellt und besonders im Hinblick auf eine Implementation in einem relationalen DBMS und den speziellen Anforderungen von ATKIS diskutiert.

### 4.1 Wege

Eine einfach zu realisierende Lösung für die in Abschnitt 2.5 beschriebene Anforderung (Bereitstellung aller Einfüge-, Lösch- und Änderungsoperationen für die Erzeugung der Inkrementellen Fortführungsinformation) ist die Aufzeichnung aller Veränderungsoperationen am ATKIS-Datenbestand (Logging-Verfahren). Das versetzt den Anwender in die Lage, seine Daten durch die Einarbeitung dieser Fortführungsinformationen immer auf dem aktuellen Stand zu halten. Er ist aber verpflichtet, die Änderungen *jeder* Fortführung in der korrekten Reihenfolge einzuarbeiten. Die spätere Rekonstruktion der einmal an einen Nutzer abgegebenen Daten wird durch ein solches Verfahren wenn nicht unmöglich gemacht, so doch extrem kompliziert.

Der Forderung nach Rekonstruierbarkeit kann man dadurch gerecht werden, daß nach jeder Fortführung ein Schnappschuß der ATKIS-Daten aufgehoben wird (Schnappschußverfahren). Dies ist einfach zu realisieren und erlaubt die Recherche alter Daten. Die Gewinnung der inkrementellen Fortführungsinformation wird jedoch schwierig. Außerdem ist dieses Verfahren mit einem Speicheraufwand von mehreren 100 Mbyte pro Sicherung selbst angesichts rapide sinkender Preise für Speichermedien keine empfehlenswerte Vorgehensweise.

Eine bessere Lösung des Problem es ist die Integration der Zeit in das ATKIS-Datenmodell. Ein solches *temporales* Datenmodell ermöglicht die Abbildung der Geschichte eines Objektes, wobei jeder einmal aktuelle Objektzustand auch nach Änderungs- oder Löschoptionen in der Datenbank verbleibt. Diese Modellierung eröffnet eine Reihe von Vorteilen:

- genaue Kontrolle über Versionen von Datenbeständen,
- leichte Erzeugung von inkrementellen Fortführungsinformationen,

- Verfügbarkeit jedes einmal aktuellen Datenbestandes,
- Möglichkeit der Verfolgung von zeitlichen Entwicklungen bei Objekten,
- Möglichkeit der Korrektur älterer Objektzustände ohne Zerstörung des fehlerhaften, aber schon an Dritte abgegebenen Zustandes.
- geringer Speicherplatzbedarf

Voraussetzung dafür ist ein temporales Datenmodell, das Einträge und Abfragen bezüglich zweier Zeitebenen unterstützt, der Zeit, zu der ein Datensatz erfaßt bzw. aktualisiert wurde (Aufzeichnungszeit) und der Zeitspanne, in der seine Informationen als gültig angenommen werden (Gültigkeitszeit). Als Hauptnachteile eröffnen sich der im Vergleich zu den beiden erstgenannten Verfahren höhere Aufwand und der höhere Bedarf an aktivem Speicherplatz. Der letztgenannte Punkt läßt sich jedoch durch die Auslagerung „historischer“ Daten entschärfen.

Ein anderes Problem bei der Fortführung der ATKIS-Daten eröffnet sich durch die anhaltende Entwicklung von ATKIS. Auch in Zukunft ist mit Veränderungen der OK zu rechnen. Wir sind also auch bei der ATKIS-Datenstruktur mit temporalen Veränderungen konfrontiert. Die Aspekte einer zeitveränderlichen Datenstruktur und der Integration der Zeit in das Metadatenystem DST werden in Abschnitt 5 besprochen.

Zunächst sollen jedoch temporale Datenmodelle vorgestellt, diskutiert und für ATKIS angewendet werden.

## 4.2 Entwicklung Temporaler Datenmodelle

Das relationale Datenmodell ist ein Konzept zur Datenstrukturierung, Integritätsdefinition und zur Manipulation und Auswertung von Daten. Die dieses Modell unterstützenden relationalen DBMS (RDBMS) verwalten Daten in Tabellen mit Zeilen und Spalten. Die Zeit kann in einem solchen herkömmlichen RDBMS nur als normales Element einer Tabelle behandelt werden, ohne dabei die Spezifika von zeitlichen Zusammenhängen abbilden zu können. So wird bei zeitlichen Abfolgen von Objektzuständen nur jeweils der aktuell gültige gespeichert. Viele Anwendungen, insbesondere aus den Bereichen Buchhaltung, Bankwesen und Statistik, aber auch Kartographie, verlangen von einem DBMS die vollständige Aufzeichnung aller jemals aktuellen Daten. Diese Anforderung führte bereits Ende der siebziger Jahre zur Entwicklung erster temporaler Datenmodelle (siehe z.B. [Ben79], [Ben82]). In den folgenden Jahren bis hin zur heutigen Zeit entstand eine Vielzahl von Veröffentlichungen zu diesem Thema. Als herausragende Autoren auf diesem Gebiet sind z.B. Ch. S.

Jensen, J. Clifford und R. T. Snodgrass zu nennen (z.B. [Tan93], [Cli86], [Sno90]). Es entstanden Konzepte zur Erweiterung des relationalen Modells um die temporale Komponente, zur Schaffung einer temporalen Abfragesprache (zum Beispiel TSQL, HQuel, TOSQL) bis hin zur Integration der Zeit in objektorientierte Systeme. Trotz des langen Forschungszeitraumes ist die praktische Umsetzung bisher über einige Prototypimplementationen nicht hinausgekommen.

## 4.3 Temporale Grundbegriffe

### 4.3.1 Zeit

Das charakteristische Element aller temporalen Daten ist die Zeit. Für unsere Belange ist es sinnvoll, von einer für alle betrachteten Objekte gleichförmig und kontinuierlich fortschreitenden Zeit auszugehen.

Wir nehmen für die Messung der Zeit ein diskretes Modell an, das heißt wir betrachten die Zeit als isomorph zur Menge der natürlichen Zahlen. Dazu wird die eigentlich stetig zu betrachtende Zeit in gleichlange Zeitintervalle eingeteilt. Man spricht bei diesem Prozeß auch von der *Granularisierung* der Zeit. Eine exakte Definition des Begriffes Granularität nimmt R. Snodgrass in [Sno95/1] vor:

**Definition 4.1 (Granularität)** *Eine Granularität wird bestimmt durch eine Länge und einen Ausgangspunkt. Die Länge wie auch der Ausgangspunkt werden in Chronons der zugrundeliegenden Uhr angegeben.*

Dabei ist ein Chronon die kleinste, im gegebenen Zusammenhang sinnvoll zu unterscheidende Zeiteinheit. In Bezug auf Datenbanksysteme ist oft die Sekunde, manchmal auch die Stunde oder der Tag ein hinreichend genaues Chronon.

Diese Sicht auf Granularitäten geht von einer Unterteilung der Zeit in gleich lange Intervalle, auch *Granulate* genannt, aus. Sie soll im weiteren *konstante Granularisierung* genannt werden. Sie ist für viele Anwendungen sinnvoll und hinreichend. Andere Probleme lassen sich aber besser mittels variabler Zeitintervalle beschreiben. So können zum Beispiel Versionen von Metadaten sinnvoller durch Zeitgranulate, die genau den Abständen zwischen Änderungen der Metadaten entsprechen, beschrieben werden. Diese Art der Unterteilung soll in Zukunft als *variable Granularisierung* bezeichnet werden.

Durch jede dieser Granularisierungen wird unsere lineare Zeitachse in sich nicht überlappende, den betrachteten Abschnitt der Geschichte aber vollständig abdeckende Stücke geteilt.

Es sollen nun aufbauend auf diesen Voraussetzungen einige weitere Begriffe definiert werden.

**Definition 4.2 (technischer Zeitpunkt)** *Im Unterschied zum eigentlichen Sinne des Wortes beschreibt ein technischer Zeitpunkt nicht einen konkreten Punkt, sondern einen Abschnitt der Zeitachse entsprechend der betrachteten Granularität. Dieser technische Zeitpunkt wird aus einem natürlichen Zeitpunkt durch Abbildung auf den entsprechenden Abschnitt der betrachteten Granularität gewonnen.*

Im weiteren ist mit einem Zeitpunkt immer der technische Zeitpunkt gemeint.

**Definition 4.3 (Zeitintervall)** *Ein Zeitintervall ist die Abbildung einer natürlichen Zeitspanne auf die betrachtete Granularisierung der Zeit. Dabei wird der Anfangs- und Endpunkt der Zeitspanne auf die entsprechenden Zeitpunkte abgebildet.*

**Definition 4.4 (Zeitdauer)** *Die Zeitdauer eines Zeitintervalls im technischen Sinne ist die Differenz von Anfangs- und Endpunkt des Intervalls in Chronons. Dabei wird für den Anfangspunkt der Beginn und für den Endpunkt das Ende des zugeordneten Zeitabschnittes auf dem Zeitstrahl gerechnet. Bei konstanter Granularität kann die Zeitdauer auch in Granulaten gerechnet werden.*

Mit diesen Voraussetzungen macht es nun Sinn, verschiedene aus Datenbanksicht relevante Zeitpunkte und Zeitintervalle zu betrachten.

**Definition 4.5 (Gültigkeitszeit  $t_g$ )** *Die Gültigkeitszeit einer Aussage ist der Zeitpunkt oder das Zeitintervall, in dem die Aussage wahr ist bzw. als wahr angesehen wird. Die Gültigkeitszeit kann in eine oder beide Richtungen unbeschränkt sein.*

**Definition 4.6 (Aufzeichnungszeit  $t_a$ )** *Die Aufzeichnungszeit einer Aussage ist der Zeitpunkt, zu dem die Aussage in ein Datenbanksystem eingebracht wurde.*

**Definition 4.7 (Beobachtungszeit  $t_b$ )** *Die Beobachtungszeit einer Aussage ist der Zeitpunkt, zu dem die Richtigkeit der Aussage erkannt wurde.*

Die einzige feststehende Aussage über die Ordnung dieser Zeiten ist  $t_b < t_a$ , ansonsten können Gültigkeits-, Aufzeichnungs- und Beobachtungszeit in beliebigem Zusammenhang stehen.

In der Praxis haben vor allem Modelle Bedeutung, die die Aufzeichnungszeit (*Aufzeichnungszeitmodelle*, 3 bekannte Ansätze), die Gültigkeitszeit (*Gültigkeitszeitmodelle*, 14 bekannte Ansätze) oder beide (*Bitemporale Modelle*, 6 bekannte Modelle) unterstützen. Die Angaben über die Verwendung der Zeitmodelle sind [Jen93/2] entnommen, das eine Übersicht der bedeutendsten temporalen Datenmodelle enthält.

Später wird auch noch der Begriff des *gesicherten Zeitpunktes* wichtig sein. Das soll ein Zeitpunkt sein, an dem eine bestimmte Information als sicher gilt, z.B. weil dies genau der Zeitpunkt ihrer Messung oder Beobachtung ist. Im Gegensatz dazu kann man eine Aussage über die Information zu einem *unsicheren Zeitpunkt* nur mit einer gewissen Wahrscheinlichkeit treffen. Diese Begriffe sind vor allem im Zusammenhang mit schnell veränderlichen Daten von Bedeutung. Sie haben aber nichts mit der Genauigkeit des Zeitpunktes selbst zu tun. In allen Überlegungen gehen wir davon aus, daß im Rahmen der gewählten Granularisierung die exakte Zeit immer bekannt ist und keinen Unsicherheiten unterliegt.

### 4.3.2 Geschichten

Bereits 1984 führt Härder den Begriff der Geschichte ein. Er wird dort wie folgt beschrieben:

**Definition 4.8 (Geschichte)** *Eine Geschichte ist eine zeitlich geordnete Folge von Zuständen oder Ereignissen von Aussagen, die zu einem Objekt oder einer Beziehung gehören. [Här84]*

Ein Objekt kann dabei ein einzelnes Attribut, eine Tabellenzeile oder auch eine komplexe, über mehrere Tabellen verteilte Struktur sein. Der Einfachheit halber werden für die weiteren Betrachtungen sowohl Objekte als auch Beziehungen als Objekte bezeichnet. Der Geschichte eines Objektes werden zeitlich geordnet alle oder einige Zustände oder vermeintliche Zustände innerhalb des betrachteten Zeitintervalls zugeordnet. Dieses Zeitintervall nennt man auch Lebenszeit eines Objektes.

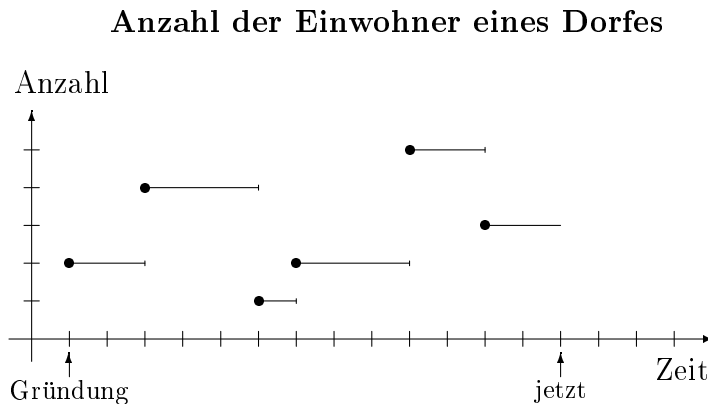


Abbildung 10: Beispiel für eine zustandserhaltende Geschichte

Es können mehrere Arten von Geschichten unterschieden werden. Die am häufigsten anzutreffende Art ist die sogenannte *zustandserhaltende Geschichte*. Hierbei wird davon ausgegangen, das ein Objekt über einen gewissen Zeitraum unveränderlich bleibt und sich



bei Eintreten eines Ereignisses abrupt ändert. Die Art der Geschichte läßt sich durch Aufzeichnung der Aussagen, die jeweils einen neuen Zustand eines Objektes beschreiben, erhalten [Här86]. Ein solcher Zustand wird auch *Version* eines Objektes genannt (siehe dazu Abschnitt 4.5). Eine exakte Definition könnte etwa wie folgt lauten:

**Definition 4.9 (Zustandserhaltende Geschichte)** *Eine zustandserhaltende Geschichte ist eine Geschichte von Aussagen, die jeweils einen neuen Zustand eines Objektes beschreibt. Ein Zustand bleibt bis zum Inkrafttreten eines neuen Zustandes invariant.*

Aus dieser Definition läßt sich ableiten, daß es bei dieser Art von Geschichte in der Lebenszeit eines Objektes keine Löcher und keine Überlappungen geben kann, das heißt zu jedem beliebigen Zeitpunkt ist der Zustand des Objektes eindeutig definiert. Ein Beispiel für eine zustandserhaltende Geschichte ist in Abbildung 10 veranschaulicht.

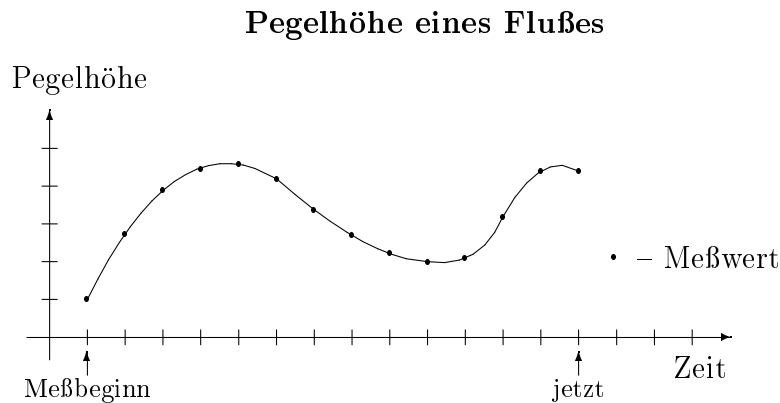


Abbildung 11: Beispiel für eine zustandsverändernde Geschichte

Nun gibt es aber in der Praxis verschiedene Anwendungen, bei denen das vorgestellte Modell nicht greift, weil sich die beschriebenen Objekte kontinuierlich verändern. Beispiele hierfür sind Temperatur-, Druck und Strahlungsverläufe. Theoretisch müßte, um ein solches Phänomen exakt aufzuzeichnen, zu jedem Zeitpunkt ein Wert abgelegt werden. Mit der Wahl einer geeigneten Granularität ist das auch in einigen Fällen möglich. Bei den meisten Anwendungen wird man jedoch mit weniger Werten auskommen müssen. Um dennoch für jeden Zeitpunkt der Lebenszeit eines Objektes einen definierten Wert zu erhalten, können zum Beispiel Interpolationsverfahren zur Anwendung kommen. Eine solche Geschichte nennt man *zustandsverändernde Geschichte* (siehe Beispiel in Abb. 11).

**Definition 4.10 (Zustandsverändernde Geschichte)** *Eine zustandsverändernde Geschichte ist eine Geschichte von kontinuierlich veränderlichen Objektzuständen. Der funktionale Zusammenhang von Zeit und Zustand ist stetig, aber unbekannt.*

Die Erfassung einer zustandsverändernden Geschichte kann in der Praxis exemplarisch durch Zustände hinreichender zeitlicher Dichte erfolgen.

Sehr ähnlich verhält es sich bei der *ableitbaren Geschichte*. Auch hier werden nur einzelne Zustände eines sich kontinuierlich verändernden Objektes erfaßt, die Zustände zwischen diesen Zeitpunkten können jedoch durch funktionale Zusammenhänge eindeutig bestimmt werden. Diese Art von Geschichte ergibt sich beispielsweise für ein Girokonto mit Soll- und Habenzinsen (siehe Abb. 12). Hier nun eine etwas exaktere Einführung dieses Begriffes:

**Definition 4.11 (Ableitbare Geschichte)** *Eine ableitbare Geschichte ist eine Geschichte von Objektzuständen, die sich an einigen Zeitpunkten sprunghaft und zwischen diesen kontinuierlich verändern. Die kontinuierliche Änderung folgt einem bekannten funktionalen Zusammenhang.*

### Kontoentwicklung mit Soll- und Habenzinsen

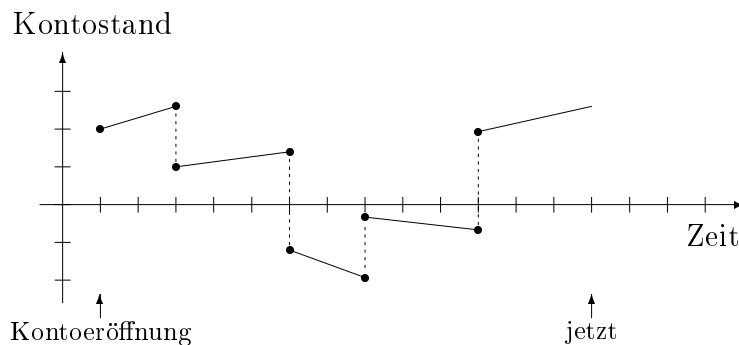


Abbildung 12: Beispiel für eine ableitbare Geschichte

Kommen wir nun zur letzten hier betrachteten Geschichtsart. Die *ereignisorientierte Geschichte* beschreibt Objekte, welche nur zu bestimmten Zeitpunkten einen definierten Wert annehmen; zwischenzeitlich sind sie undefiniert. Dabei ist es auch denkbar, daß zu einem Zeitpunkt mehrere Werte gültig sind. Eine solche Geschichte würde beispielsweise die Baumentnahme aus einem bestimmten Waldstück ergeben (siehe Abb. 13). Auch hierzu soll die entsprechende Definition angegeben werden.

**Definition 4.12 (Ereignisorientierte Geschichte)** *Eine ereignisorientierte Geschichte ist eine Geschichte von Objektzuständen, die nur zu gewissen Zeitpunkten definiert sind. Es kann Zeitpunkte ohne und Zeitpunkte mit mehreren Objektzuständen geben.*

Abschließend sei noch angemerkt, daß in einigen Literaturstellen zustandserhaltende Geschichten auch als zustandsorientierte oder zustandsstabile und zustandsverändernde als kontinuierliche Geschichten bezeichnet werden.

## Baumentnahme aus einem Wald

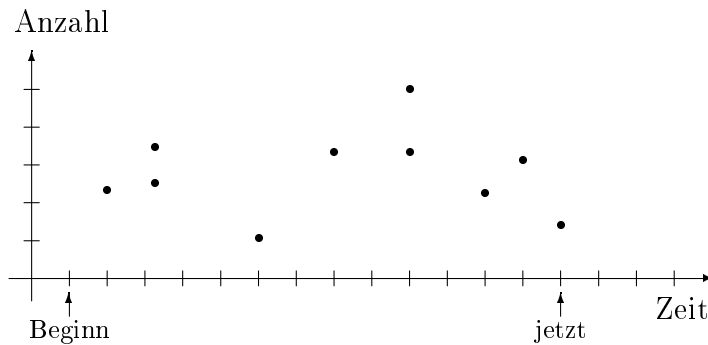


Abbildung 13: Beispiel für eine ereignisorientierte Geschichte

### 4.4 Geschichten in ATKIS

#### 4.4.1 Geschichten spatialer Daten

Mit welcher dieser Geschichtsarten läßt sich nun die Entwicklung der ATKIS-Objekte und -Objektteile am besten beschreiben? Sehen wir uns zunächst die Veränderungen auf der geometrischen Ebene an. In der Natur beobachten wir sowohl kontinuierliche Veränderungen, zum Beispiel das Vorrücken einer Steilküste ins Landesinnere, als auch abrupte, wie die Erweiterung der Grundfläche eines Gebäudes durch Anbau.

Der erste Fall eröffnet sofort gewisse Schwierigkeiten. Aufgrund des enormen technischen Aufwandes ist es in der heutigen Praxis nicht möglich, mehr als Schnappschüsse von der Oberflächenstruktur der Erde zu erhalten. Zudem sind die funktionalen Zusammenhänge kontinuierlicher Lageänderungen im allgemeinen nicht bekannt. Man kann also davon ausgehen, daß solche Veränderungen nur durch eine zustandsverändernde Geschichte beschrieben werden können.

Wie bereits in 4.3.2 beschrieben, geht man bei dieser Art von Geschichte davon aus, daß man Zustände zwischen den gesicherten Zeitpunkten mit vertretbarer Genauigkeit interpolieren kann. Zur Zeit sind die Abstände zwischen den Fortführungsphasen topographischer Datenbestände noch sehr groß, eine Simulation der Entwicklung der Geometrie eines topographischen Objektes muß aber nicht zwingend zu ungenau werden, da solche Veränderungen im allgemeinen eher langsam vonstatten gehen. Nehmen wir zum Beispiel den Verlauf eines Flusses. Seine Änderungen werden im Laufe eines Jahres nur wenige Meter betragen. So erhält man selbst bei Fortführungszeiträumen von einigen Jahren ein hinreichend genaues Bild der Flußveränderungen.

Im zweiten genannten Fall handelt es sich um eine zustandserhaltende Geschichte, wobei der Zeitpunkt der tatsächlichen Änderung im allgemeinen nicht bekannt ist. Entgegen 4.3.2

ist es also hier nicht möglich, zu jedem Zeitpunkt eine Aussage zur Geometrie des Objektes zu treffen. Ein korrekter Objektzustand ist nur für den Zeitpunkt der Erfassung bzw. Fortführung vorauszusetzen.

Leider ist es nicht möglich, die Art der Geschichte aus der Art des abgebildeten Objektes zu folgern. So kann sich die Lage des als Beispiel genannten Flusses auch plötzlich durch Eingriffe von Menschenhand oder Naturkatastrophen verändern.

Eine mögliche Lösung ist das Verzichten auf zustandsverändernde Geschichten und die Abbildung aller temporalen Vorgänge mittels einer zustandserhaltenden Geschichte. Eine ereignisorientierte Geschichte, wie sie sich scheinbar anbietet, ist nicht ausreichend, da zwischen den gesicherten Zeitpunkten sehr wohl ein Objektzustand existiert. Man muß sich allerdings bewußt sein, daß dieser, obwohl bei zustandserhaltender Geschichte immer definiert, nur mit einer gewissen Wahrscheinlichkeit korrekt ist. Wie bereits besprochen, wird er aber in den meisten Fällen eine gute Annäherung an den tatsächlichen Zustand sein.

#### **4.4.2 Geschichten aspatialer Daten**

Wenden wir uns nun der Modellierung temporaler Entwicklungen der attributiven ATKIS-Daten zu. Hier sind wieder verschiedene Arten von Geschichten anzutreffen. Viele ATKIS-Attribute beschreiben topographisch relevante Sachverhalte, indem die hervorstechendsten Eigenschaften eines Objektes in möglichst treffenden Kategorien eingeordnet werden. Diese Abstraktion von ansonsten nahezu unbegrenzt vielfältigen Charakteristika macht erst eine sinnvolle Verarbeitung und vergleichende Auswertung topographischer Objekte möglich. Sie verhindert aber auf der anderen Seite die Dokumentation stetig verlaufender Veränderungen in der Landschaft. So läßt sich die allmähliche Verwandlung eines Laubwaldes in einen Mischwald nur durch die Änderung des Attributes Waldtyp (VEG) festhalten, wenn ein bestimmter Nadelbaumanteil überschritten wurde. Der zeitliche Verlauf der „Vernadelung“ kann nicht nachvollzogen werden. Wir beobachten hier also wiederum eine zustandserhaltende Geschichtsmodellierung, die dem Wesen nach eigentlich eine zustandsverändernde ist.

Etwas anders liegt der Fall bei Attributen, die nicht kodierte Eigenschaften von topographischen Objekten, sondern echte Maßangaben beinhalten. Dabei handelt es sich um Breiten- und Höheninformationen, die nicht direkt aus der Geometrie ablesbar sind, die Tragfähigkeiten von Bauwerken, die Spannung eines Umspannwerkes und vieles andere. Einige dieser Attribute sind von Natur aus diskret, wie zum Beispiel die Anzahl der Startbahnen eines Flughafens oder die Anzahl der Geschosse eines Gebäudes. Andere, wie die mittlere Pegelhöhe eines Gewässers, haben einen stetigen Charakter.

Wir beobachten hier eine ähnlich gelagerte Situation wie bei den spatialen ATKIS-Daten. Für Teilaspekte wäre eine zustandsverändernde Geschichte wünschenswert. Die Vor-

aussetzungen für eine relativ genaue Interpolation sind auch hier trotz der großen Abstände gesicherter Zeitpunkte positiv zu bewerten, da sich die betroffenen Attributwerte meist nur in längeren Zeiträumen wesentlich verändern. Der größere Teil der Attribute unterliegt aber einer zustandserhaltenden Geschichte.

Es ergeben sich somit zwei Möglichkeiten, die Erschaffung einer komplexen Geschichtsverwaltung mit mehreren Geschichtsarten oder die vereinfachte Modellierung nur mit zustandserhaltenden Geschichten und einem wesentlich geringerem technischen Aufwand.

Wir müssen nun abwägen, welcher dieser beiden Wege eher dem Wesen und den Anforderungen von ATKIS entspricht. Eine die wahre Natur vieler topographischer Objekte besser beschreibende Lösung mit verschiedenen Arten von Geschichten ist ohne Zweifel im Sinne einiger Anwendergruppen wie Ökologen, Historiker und Umweltschützer, die vor allen Dingen an Entwicklungen der Landschaft interessiert sind. Das Gros der ATKIS-Benutzer ist aber eher im kartographischen, topographischen und raum- und landschaftsplanerischen Umfeld angesiedelt. Für solche Anwender ist in erster Linie eine Abbildung der Landschaft zu einem konkreten Zeitpunkt wichtig, weniger aber die temporale Entwicklung einzelner Objekte. Es gilt also, die Recherchierbarkeit des Zustandes des DLM zu einem beliebigen Zeitpunkt sicherzustellen, wobei für den aktuellen Stand der einfachste Zugriff gewährleistet sein sollte.

Zusammenfassend ist festzustellen, daß für alle in ATKIS verwalteten Informationen, die temporalen Änderungen unterworfen sind, die Modellierung mit Hilfe einer zustandserhaltenden Geschichte akzeptabel ist.

## 4.5 Versionen

Da wir im Rahmen von ATKIS offensichtlich hauptsächlich mit zustandserhaltenden Geschichten konfrontiert werden, erweist es sich als notwendig, den in Abschnitt 4.3.2 nur kurz erwähnten Versionenbegriff näher zu beleuchten.

In der älteren Literatur wird die Version etwa in folgendem Sinne eingeführt:

**Definition 4.13 (Version(1))** *Eine Version ist ein Zustand eines Objektes, der über ein gewisses, Gültigkeitszeit der Version genanntes Zeitintervall einer zustandserhaltenden Geschichte unveränderlich ist. Die den Versionen eines Objektes zugeordneten Zeitintervalle überschneiden sich nicht und bilden zusammengefaßt den unterbrechungsfreien Lebenszeitraum des Objektes. Eine neue Version eines Objektes wird durch ein Ereignis erzeugt.*

Diese Definition hat unter anderem zur Folge, daß ein Objekt während seiner Lebenszeit immer einen definierten Zustand besitzt. Lücken in der Objektgeschichte werden ausgeschlossen. Wir betrachten auch in der realen Welt ein Objekt, das für eine gewisse Zeitdauer nicht

mehr existiert hat, in der Regel als ein neues Objekt. Ein Gebäude, das abgerissen und wieder aufgebaut wurde, wird nicht als ein und derselbe Gegenstand betrachtet werden, auch wenn eine starke Korrelation zwischen beiden besteht.

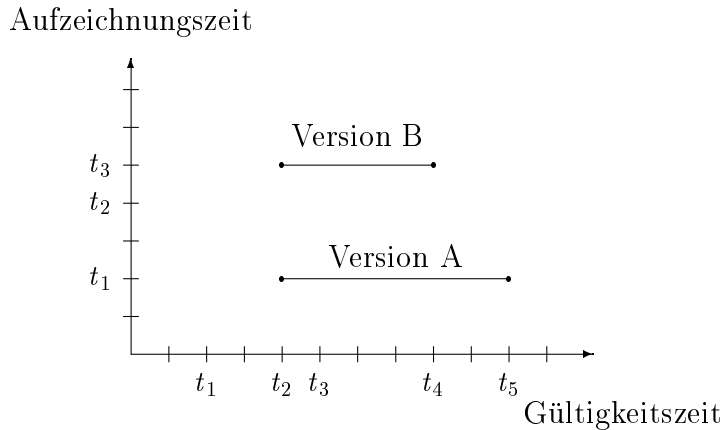


Abbildung 14: Beispiel für eine bitemporale Objektgeschichte

Eine andere Konsequenz dieser Definition ist die strikte Linearität einer Objektgeschichte. Bei ATKIS werden wir jedoch mit einer *bitemporalen Objektgeschichte* konfrontiert. Nehmen wir zum Beispiel den in Abbildung 14 dargestellten Fall. Zum Zeitpunkt  $t_1$  wird die Version *A* eines Objektes gespeichert. Sie soll einen Gültigkeitszeitraum von  $t_2$  bis  $t_5$  haben. Zu einem Zeitpunkt  $t_3 > t_1$  stellt man jedoch fest, dass die Version *A* für den Zeitraum  $t_2$  bis  $t_4$  fehlerhaft ist. Nun wurde die Version *A* aber bereits an andere Nutzer weitergegeben. Eine Änderung der Version *A* in die korrekte Version *B* würde zwar die Linearität der Objektgeschichte erhalten, die Recherchierbarkeit der wahren Objektgeschichte aber unmöglich machen. Also wird eine neue Version *B* erzeugt.

Wir müssen also sowohl den Aufzeichnungszeitpunkt  $t_a$  als auch das Gültigkeitszeitintervall  $t_g$  eines Objektes betrachten. Eine Version eines Objektes existiert dann immer bezüglich des Tupels  $(t_a, t_g)$ . Bezüglich einem der beiden Zeitstrahlen können also mehrere Versionen eines Objektes existieren.

Aus diesen Überlegungen heraus wird in der moderneren Literatur (siehe z.B. [Kim90], [Sno95/2] und [Cat91]) die Version eines Objektes etwa wie folgt eingeführt:

**Definition 4.14 (Version(2))** *Eine Version ist ein Zustand eines Objektes. Sie besitzt einen Gültigkeitszeitraum und ist je nach Art der Version unveränderlich oder veränderlich. Nach der Erzeugung einer ersten Version werden alle weiteren Versionen eines Objektes von dieser rekursiv abgeleitet und bilden eine Versionenhierarchie.*

Dadurch ergibt sich ein Versionenschema wie in Abbildung 15 dargestellt. Aus objektorientierter Sicht sind alle Versionen Instanzen eines Objektes. In einem relationalen Modell sind das Objekt und seine Versionen durch eine  $1 : n$ -Beziehung verbunden.

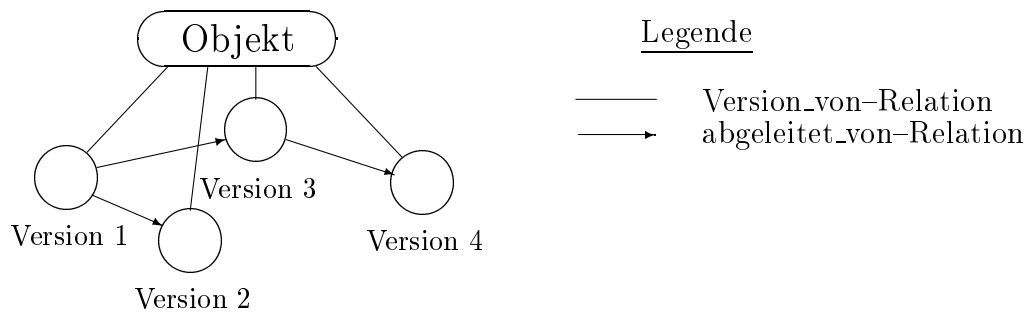


Abbildung 15: Beispiel einer Versionenhierarchie [Kim90]

Wie in der Definition bereits angedeutet, werden mehrere Arten von Versionen unterschieden. Diese Notwendigkeit ergibt sich vor allem aus der praktischen Arbeit mit temporalen Daten. Eine der Grundeigenschaften temporaler Daten ist ihre unbegrenzte Dauerhaftigkeit, das heißt sie werden, einmal in eine Datenbank eingebracht, nie wieder verändert oder gar gelöscht. Soll ein Objekt verändert werden, so ist eine neue Version des Objektes zu erzeugen. Nun befinden sich aber gerade im GIS-Bereich Daten während eines Bearbeitungszeitraumes oft in einem unfertigen Zustand, der noch nicht dem erwünschten Endzustand entspricht. Wir benötigen also für die praktische Handhabung Versionenzustände, die noch Änderungen oder gar Löschungen zulassen, ohne das eine neue Version erzeugt und die alte in der Datenbank belassen werden muß.

Won Kim unterscheidet in [Kim90] drei Arten von Versionen, die sich durch den Grad ihrer *Robustheit* unterscheiden. Unter Robustheit ist hier ein Maß für die Unveränderlichkeit eines Datensatzes zu verstehen

**Definition 4.15 (Flüchtige Version (transient version))** *Eine flüchtige Version eines Objektes ist eine temporäre Version, die geändert und gelöscht werden kann. Für das Objekt geforderten Integritätsbedingungen können bei flüchtigen Versionen für ungültig erklärt werden.*

Eine flüchtige Version besitzt von allen die geringste Robustheit. Das heißt, sie kann nahezu ohne Beschränkungen verändert werden. Einen höheren Robustheitsgrad besitzt dagegen die *Arbeitsversion*.

**Definition 4.16 (Arbeitsversion (working version))** *Eine Arbeitsversion eines Objektes kann gelöscht, aber nicht geändert werden.*

Abschließend betrachten wir Versionen mit dem höchsten Robustheitsgrad, die *freigegebene Version*.

**Definition 4.17 (Freigegebene Version (released version))** *Eine freigegebene Version eines Objektes kann weder geändert noch gelöscht werden.*

Eine neu erzeugte Version eines Objektes ist immer eine flüchtige Version. Eine flüchtige Version wird durch *Beförderung* (promotion) zu einer Arbeitsversion und durch eine weitere Beförderung zu einer freigegebenen Version. Eine Arbeitsversion kann durch *Degradierung* (demote) wieder zur flüchtigen Version werden. Eine freigegebene Version kann jedoch nicht degradiert werden.

Diese Unterscheidung von Versionen ist auch für ATKIS-Daten sinnvoll. Ein ATKIS-Objekt oder -Objektteil durchläuft in seiner Entwicklung mehrere Stadien. Während seiner Erfassung oder Erzeugung durch automatische Verfahren kann es sich in einem inkonsistenten Zustand befinden, bestimmte Komponenten werden evtl. erst zu einem späteren Zeitpunkt vervollständigt oder auf ihre Richtigkeit überprüft<sup>1</sup>. Diese flüchtige Version geht mit der Beendigung der Erfassung durch Beförderung in eine Arbeitsversion über. Alle Integritätsbedingungen müssen nun erfüllt sein. An die eigentliche Erfassung schließt sich ein Prozeß der inhaltlichen Überprüfung der ATKIS-Daten an. Hier können Objekte und Objektteile noch geändert oder gelöscht werden. Für eine Änderung muß die Arbeitsversion zuerst zu einer flüchtigen Version degradiert werden, eine Löschung kann direkt erfolgen. Nach diesem letzten Arbeitsschritt werden die Daten freigegeben und können an Dritte abgegeben werden. Sie dürfen nun nicht mehr verändert oder gelöscht werden; der Erhalt von an Dritte abgegebenen Daten muß immer garantiert sein.

## 4.6 Versionen im relationalen Modell

Für die eingeführten Arten von Versionen müssen nun Repräsentationen im relationalen Modell einschließlich der Operationen Erzeugen, Löschen, Ändern und Abfrage definiert werden. Hierzu gibt es eine Reihe von Ansätzen. Eine große Anzahl von Arbeiten zu diesem Thema sieht nur eine Abbildung der Gültigkeitszeit vor, während andere Modelle nur die Aufzeichnungszeit berücksichtigen. Diese Lösungen sind für ATKIS nicht ausreichend. Wie bereits in 2.5 begründet, benötigt ATKIS die Abbildung von bitemporalen Geschichten.

Das erste bitemporale Modell wurde bereits 1982 von Ben-Zvi unter dem Namen Time Relational Model in [Ben82] beschrieben. Die Komplexität der Materie brachte noch eine Reihe von sehr unterschiedlichen Ansätzen hervor. Für ATKIS müssen wir uns für eine Variante entscheiden, die die folgenden Bedingungen erfüllt:

- Erweiterung des Relationenmodells unter Wahrung der Einfachheit (bitemporal erweitertes Relationenmodell).

---

<sup>1</sup> Ein typisches Beispiel hierfür ist die automatische Erzeugung von Objektteilen, die erst im Laufe einer späteren manuellen Bearbeitung Objekten zugeordnet werden.



- Anordnung der Zeit auf Tupelebene. Die Unterstützung temporaler Attribute ist für ATKIS nicht sinnvoll, da die Anzahl der Attribute pro Tupel i. a. gering ist, der technologische Aufwand zur Unterstützung temporaler Attribute jedoch sehr hoch.
- Bevorzugung der jüngsten Version eines Objektes (bezüglich Gültigkeits- und Aufzeichnungszeit). Getrennte Speicherung der jüngsten und älteren Versionen. Der Zugriff auf aktuelle Daten muß gegenüber dem wesentlich selteneren Zugriff auf ältere Daten bevorzugt werden.
- Unterstützung von Versionen von Metadaten (Strukturversionen) für nicht-aktuelle Daten. Die aktuellen Daten liegen per Definition immer in einer einheitlichen Struktur vor.
- Unterstützung von Anfragen bezüglich eines bitemporalen Zeitpunktes (Snapshot- oder AS OF-Anfrage) und bezüglich eines Intervalls der Gültigkeits- oder Aufzeichnungszeit (WALK THRU TIME-Anfrage).
- Unterstützung von temporalen Joins (Selektion von Geschichten aufgrund gegenseitiger zeitlicher Abhängigkeiten) unter besonderer Berücksichtigung eines temporalen Equi Join (Verbund zweier Geschichten, wenn die Verbundbedingung zur gleichen Zeit in beiden Geschichten gültig ist). Temporale Joins müssen bezüglich einer oder beider Zeitebenen definiert werden können.

#### 4.6.1 Unterscheidung bitemporal erweiterter Relationenmodelle

Die Geschichte temporaler Datenmodelle brachte bereits eine Reihe von bitemporalen Ansätzen auf Basis des Relationenmodells hervor. Diese unterscheiden sich in einigen wesentlichen Punkten, welche im folgenden kurz erläutert und im Hinblick auf ihre Relevanz für ATKIS ausgewertet werden.

Der erste wichtige Unterschied ist die Art und Weise, wie Zeitstempel dargestellt werden. Ein Zeitstempel ist dabei die Kombination aus Zeitpunkten und -intervallen, die für jede Version eines Objektes gespeichert werden. Diese Zeitstempel können als normale Attribute, deren Semantik dem RDBMS nicht bekannt ist, oder als neue Datentypen einschließlich der entsprechenden Operationen abgebildet werden. Die letzte Variante verlangt tiefere Eingriffe in das RDBMS und ist deshalb für ATKIS nicht empfehlenswert.

Ein weiterer Punkt ist die Frage, ob sich die temporalen Relationen in erster Normalform befinden. Wenn ja, dann muß eine Objektgeschichte durch eine Menge von Tupeln dargestellt werden, wobei der Zeitstempel zum Primärschlüssel der Relation hinzugenommen werden muß. Dieser Ansatz ist konzeptionell sehr nahe am relationalen Datenmodell, bringt aber

eine gesteigerte Auswertungskomplexität mit sich, da Tupel nicht geordnet sind. Im anderen Fall kann die gesamte Objektgeschichte in einem Tupel abgelegt werden. Dadurch ergeben sich Relationen folgender Struktur:

$REL(\underline{A_1, \dots, A_n}, A_{n+1}, \dots, A_k, H(TS, A_{k+1}, \dots, A_l))$

$A_1, \dots, A_n$ : Primärschlüssel der Relation

$A_{n+1}, \dots, A_k$ : nicht-temporale Attribute der Relation

$H(TS, A_{k+1}, \dots, A_l)$ : Wiederholungsgruppe aller zeitveränderlichen Aussagen der Geschichte der Relation mit dem Zeitstempel TS

Dies vereinfacht entscheidend Anfragen nach Objektgeschichten (Walk Thru Time-Query), insbesondere Anfragen über mehrere temporale Relationen (temporal join).

Eng verbunden damit ist die Frage, ob die Zeit auf Relationen-, Tupel- oder Attributebene angeordnet werden sollte. Eine Anordnung auf Relationenebene ist offensichtlich ungünstig, da bei jeder Änderung eines Tupels die gesamte Relation dupliziert werden müßte. Bei Anordnung auf Tupelebene ergibt sich in Abhängigkeit von der Normalform eines der folgenden Schemen:

NF<sup>2</sup> :  $REL(\underline{A_1, \dots, A_n}, A_{n+1}, \dots, A_k, H(TS, A_{k+1}, \dots, A_l))$

1NF:  $REL(\underline{TS, A_1, \dots, A_n}, A_{n+1}, \dots, A_k, A_{k+1}, \dots, A_l)$

Für ATKIS sollte auf eine dieser beiden Varianten zurückgegriffen werden. Der dritte Weg verlangt die Führung eines Zeitstempels für jedes temporale Attribut der Relation:

$REL(\underline{A_1, \dots, A_n}, A_{n+1}, \dots, A_k, H_1(TS, A_{k+1}), \dots, H_m(TS, A_l))$

Diese Möglichkeit ist für ATKIS aufgrund des hohen Aufwandes nicht diskutabel.

Ein weiterer bedeutender Unterschied zwischen den verschiedenen temporal erweiterten Relationenmodellen ist die Frage der Bevorzugung aktueller vor historischen Tupeln und eine getrennte Speicherung dieser. Eine solche Vorgehensweise ist von großem Vorteil bei Systemen wie ATKIS, die bevorzugt mit aktuellen Daten arbeiten. Der Zugriff auf diese wird beschleunigt und es eröffnet sich die Möglichkeit, ältere Daten zu komprimieren oder auf andere Speichermedien auszulagern. Das Führen von Versionen von Metadaten (Strukturversionen) wird ermöglicht. Allerdings werden Anfragen nach Geschichten von Objekten erschwert.

Das letzte Unterscheidungsmerkmal ist die Art der Darstellung von Zeitstempeln. Für die Gültigkeitszeit sind einzelne Zeitpunkte, offene, halboffene oder geschlossene Zeitintervalle oder ein Gültigkeitszeitelement (Menge von Zeitpunkten oder -intervallen) denkbar, die Aufzeichnungszeit kann als einzelner Zeitpunkt (now), als Zeitintervall (now, until changed) oder als drei Zeitpunkte (Aufzeichnungszeitpunkte der Gültigkeitszeitpunkte und der Todeszeit<sup>2</sup>).

---

<sup>2</sup> Mit *Todeszeit* wird der Zeitpunkt bezeichnet, ab dem keine Version eines Objektes mehr gültig ist.

Für ATKIS ist die Verwendung unterschiedlicher Zeitstempel für aktuelle und historische Daten günstig. Aktuelle Tupel sind durch einen Aufzeichnungszeitpunkt und einen Gültigkeitszeitpunkt (Beginn des Gültigkeitszeitraumes) zeitlich ausreichend eingeordnet. Für historische Daten ist zusätzlich die Aufzeichnung der Löschezit und des Endzeitpunktes des Gültigkeitszeitraumes notwendig.

## 4.7 Temporales ATKIS–Datenmodell

Als Zusammenfassung der in den letzten Abschnitten diskutierten Details ergibt sich die folgende Liste von Anforderungen an ein temporales ATKIS–Datenmodell.

- Nutzung eines herkömmlichen RDBMS (z.B. Informix oder Oracle)
- bitemporales Modell
- Anordnung der Zeit auf Tupelebene
- Trennung des aktuellen Tupels von historischen Tupeln (durch Speicherung in verschiedenen Tabellen)
- 3 Robustheitsgrade für aktuelle Tupel (historische Tupel sind immer freigegebene Versionen, da sie keinen Veränderungen unterliegen dürfen)
- Hinzufügen des Zeitstempels zum Primärschlüssel bei historischen Tupeln. Bei aktuellen Tupeln genügt die Darstellung als normale Attribute.
- als Granularität können Bearbeitungszeiträume gewählt werden, die mit auf Tagen basierenden Zeitintervallen verbunden sind. Dabei sind keine in der Zukunft liegenden Zeitpunkte zulässig.
- Zeitstempel bestehen:
  - bei aktuellen Tupeln aus der Aufzeichnungszeit (Zeitpunkt) und der Gültigkeitszeit (Zeitpunkt)
  - bei historischen Tupeln aus der Aufzeichnungszeit (Zeitintervall, bestehend aus der Aufzeichnungszeit und der Löschezit, die bei nicht gelöschten Tupeln auf  $\infty$  gesetzt wird) und der Gültigkeitszeit (Zeitintervall)

## 4.8 Temporale Operationen für ATKIS

Für ein temporales ATKIS-Datenmodell, welches die vorhergehenden Bedingungen erfüllt, muß eine Reihe von Operationen definiert und implementiert werden. Dabei unterscheiden wir einerseits Anfrage- und Änderungsoperationen, andererseits Operationen über aktuellen und historischen Tupeln. Anfrageoperationen können sich natürlich auch auf aktuelle und historische Tupel beziehen. Sie sollen jedoch kein Diskussionspunkt dieser Arbeit sein, da sie sehr spezifisch an eine Implementation anzupassen sind, die bis zum jetzigen Zeitpunkt noch nicht verwirklicht ist.

Für Änderungsoperationen, an dieser Stelle ein Oberbegriff für alle Operationen, die Manipulationen am Datenbestand vornehmen, lassen sich unabhängig von einer speziellen Implementation gewisse Grundsätze festhalten.

### 4.8.1 Änderungsoperationen für aktuelle Tupel

Als Voraussetzung für andere Operationen muß zunächst die Beförderung und Degradierung von aktuellen Tupeln entsprechend der aufgestellten Regeln bereitgestellt werden. Des weiteren müssen neue Objekte erzeugt werden können. Dabei werden Gültigkeitszeit und Aufzeichnungszeit auf *now*<sup>3</sup> gesetzt. Die neuen Tupel sind flüchtige Versionen. Für bereits existierende Objekte müssen neue temporale Instanzen erschaffen werden können. Die alte Version wird zur freigegebenen Version befördert, wird historisches Tupel und bekommt als Gültigkeitsendzeit die (Gültigkeitszeit des neuen Tupels) -1 (bezüglich der Granularität). Die neue erzeugte Version ist eine flüchtige Version. Die letzte notwendige Operation ist das logische Löschen eines Objektes. Das aktuelle Tupel wird historisches Tupel und bekommt als Gültigkeitsendzeit (*now*) -1. Eine aktuelle Version existiert dann nicht mehr.

### 4.8.2 Änderungsoperationen für historische Tupel

Für historische Tupel sind nur zwei Operationen notwendig. Die erste ist das Erzeugen eines Tupels in Geschichte. Das entspricht der Änderung von historischen Tupeln. Das Tupel wird sofort freigegebene Version. Das Gültigkeitszeitintervall ist frei wählbar, darf sich aber nicht mit dem des aktuellen Tupels überschneiden und muß innerhalb der Lebenszeit des Objektes liegen. Die Aufzeichnungszeit ist *now*. Das Tupel überlagert alle Tupel mit älterer Aufzeichnungszeit. Dies ist wichtig für Anfrageoperationen.

Die zweite ist das Löschen eines Tupels in Geschichte. Die Löschezit wird *now* gesetzt. Das Löschen eines Tupels kann zu Lücken in der Objektgeschichte führen. Das ist bei Anfrageoperationen zu berücksichtigen.

---

<sup>3</sup> Mit *now* wird der aktuelle Zeitpunkt bezeichnet.

## 4.9 Zusammenfassung

In den vorangegangenen Abschnitten wurde eine kurze Einführung in die äußerst komplexe Materie der temporalen Datenmodelle gegeben und verschiedene Anwendungsmöglichkeiten für den Aufbau des ATKIS-Datenmodells diskutiert. Dabei wurde die prinzipielle Eignung bestimmter temporaler Modelle für ATKIS nachgewiesen und Wege für die Verwirklichung aufgezeigt. Dies stellt aber lediglich einen Einstieg in eine temporale ATKIS-Modellierung dar. Verschiedene Fragen der Modellierung und einer konkreten Implementation, wie z. B. die Gestaltung einer temporalen Anfragesprache, sind noch offengelassen worden und müssen noch gelöst werden, bevor die hier dargelegten Ansätze verwirklicht werden können. Dieses muß aber Anliegen weitergehender Untersuchungen sein.

## 4.10 Weiterführende Literatur

Die Recherche für diese Arbeit resultierte unter anderem in einer umfangreichen Literaturstellensammlung zum Themenbereich temporale Datenmodelle. Es handelt sich dabei um Veröffentlichungen in Buchform, in Zeitschriften oder Zeitschriftenbänden und im Internet. Letzteres können sowohl Veröffentlichungen von wissenschaftlichen Institutionen oder Tagungen als auch Internetseiten mit Bedeutung für diesen Themenkomplex sein. Die vollständige Literaturliste befindet sich im Anschluß an das normale Literaturverzeichnis. In dieser sind auch alle für diese Arbeit verwandten Literaturstellen noch einmal aufgeführt. Bei einigen Einträgen fehlen verschiedene Daten wie Erscheinungsjahr oder -ort. Dabei handelt es sich um Veröffentlichungen, die nur im Internet aufgefunden wurden und dort ohne die entsprechenden Angaben vorlagen.

Weiterhin enthält diese Arbeit im Anschluß an das Literaturverzeichnis eine Liste von Internetadressen, die auf Seiten verweisen, welche sich zum Zeitpunkt der Arbeit mit dem Problemkreis temporale Datenmodelle beschäftigten. Diese Aufstellung erhebt keinen Anspruch auf Vollständigkeit, stellt aber einen Querschnitt von Veröffentlichungen dar, der vor allem für den Einstieg in die Thematik nützlich sein kann.

## 5 Versionen von Metadaten

Die Geschichte der Geoinformationssysteme und ATKIS zeigt, daß Modelle für Geodaten selten über längere Zeiträume unveränderlich sind. Die beschreibende Struktur verändert sich durch die rasante Entwicklung der Anforderungen an Geodaten. Auch die komplizierte Modellierung und die damit oft notwendig werdenden Korrekturen an Strukturen tragen zu diesem Prozess bei. Auf der anderen Seite besteht aber nicht nur ein Bedarf an aktuellen Daten, es werden auch ältere Versionen von Geodaten benötigt. Darüber hinaus ist es notwendig, nachträgliche Korrekturen in ältere Datenbestände einzubringen, dabei aber die originalen Daten zu erhalten. All dies erfordert eine komplexe Zeitverwaltung auf Daten- und Metadatenebene. Um die Voraussetzungen dafür zu schaffen, sollen im folgenden einige Konzepte zur Verwaltung von Versionen von Metadaten (Strukturversionen) entwickelt und näher untersucht werden.

### 5.1 Allgemeine Begriffe

Zunächst sollen einige grundlegende Begriffe eingeführt werden.

**Definition 5.1 (Metadaten)** *Informationen über Struktur und Eigenschaften von Daten. Metadaten erlauben den effizienten und automatischen Zugriff auf komplexe Datenstrukturen.*

**Definition 5.2 (Version von Metadaten)** *Eine Version ist ein Zustand der Metadaten. Dabei kann eine Version über einen längeren Zeitraum gültig sein. Eine neue Version entsteht, wenn Metadaten in einer Weise verändert werden, so daß sich die Struktur oder die Eigenschaften der beschriebenen Daten ändern.*

**Definition 5.3 (Tabellentyp)** *Ein Tabellentyp ist eine Klasse von gleichartigen Tabellen, welche in gleicher Weise definierte und nach einheitlichen Namenskonventionen bestimmte Schlüssel- und Fremdschlüsselattribute enthalten, in gleicher Weise mit Tabellen anderer Typen verbunden sind (über Primär-Fremdschlüsselbeziehungen und weiterführende Verknüpfungen) und gleiche oder artgleiche technologische Attribute beinhalten (z.B. für Zeitverwaltung). Eine Tabelle eines Tabellentypes heißt Instanz dieses Tabellentypes. Informationen über Tabellentypen sind üblicherweise nicht in Metainformationssystemen verfügbar und sind somit zusätzliche Voraussetzung für eine Interpretation der Metadaten.*

**Definition 5.4 (Tabellencluster)** *Als Tabellencluster bezeichnet man eine Menge von Tabellentypen, die sich aufgrund von Primär-Fremdschlüsselbeziehungen oder anderer Verknüpfungen gegenseitig bedingen und erfordern. Eine Instanz eines Tabellenclusters ist eine*

Menge real existierender Tabellen, wobei von jedem Tabellentyp des Tabellenclusters jeweils eine Instanz existiert.

## 5.2 Metadaten von zeitveränderlichen Daten

Für die nachfolgenden Überlegungen nehmen wir an, daß nur die Metadaten veränderlich sind, die Struktur der Metadatenbank aber fest ist. Metadaten verändern sich nur bezüglich einer linearen Zeitachse, bitemporale Operationen wie Veränderungen in der Vergangenheit werden nicht unterstützt. Des weiteren unterscheiden wir zwei Arten von Veränderungen von Metadaten, die, welche die Existenz von Tabellen beginnen oder beenden, und die, die Änderungen von existierenden Tabellen nach sich ziehen. Das können Veränderungen der Spalten und der Eigenschaften von Tabellen sein.

Es wird nur der spezielle Fall betrachtet, daß sowohl eine Versionierung der Metadaten als auch der eigentlichen Daten erfolgt. Daraus ergeben sich bereits einige Einschränkungen.

Die Granularität der Zeit bezüglich der Metadaten muß in der Granularität der Zeit der beschriebenen Daten enthalten sein. Das heißt zu jedem technischen Zeitpunkt bezüglich der beschriebenen Daten darf nur eine Version der Metadaten gültig sein.

Des weiteren dürfen Tabellen zwar neu erzeugt, später jedoch nicht wieder gelöscht werden, da jede Tabelle freigegebene Versionen von Objekten beinhalten kann, die nie wieder gelöscht werden dürfen. Es ist nur eine Markierung der Tabelle als *inaktiv* möglich.

Wird eine neue Instanz eines Tabellentypes erzeugt, der Element eines Tabellenclusters ist, so müssen auch Instanzen aller anderen Tabellentypen dieses Clusters erzeugt werden.

Die folgenden Komponenten einer Tabelle dürfen sich nicht mehr verändern, sobald Daten in die Tabelle eingebracht sind:

- der Name der Tabelle,
- der Tabellentyp, also alle Primär und Fremdschlüssel sowie allgemeine Attribute für die Zeitverwaltung und evt. stets vorhandene Attribute dieses Tabellentyps inklusive aller diese Spalten betreffenden Integritätsbedingungen,
- Spalten, in denen sich Daten befinden, deren Informationsgehalt nicht vollständig in anderen Spalten enthalten ist, die also nicht redundant abgelegt sind.

Existierende Tabellen können durch die folgenden Operationen verändert werden:

- Anfügen einer Spalte,
- Erweitern, Einschränken oder neu Definieren einer Integritätsbedingung, wenn dadurch keine in dieser oder einer anderen Tabelle enthaltenen Tupel undefiniert werden,

- Löschen einer redundanten Spalte, sofern die Recherchierbarkeit historischer Daten dadurch nicht eingeschränkt wird, und
- logisches Löschen einer Spalte, in die in Zukunft keine Daten mehr eingebracht werden sollen. Jedem neuen Tupel der Tabelle wird für die logisch gelöschte Spalte ein definierter Wert mit der Bedeutung „Attribut nicht mehr belegt“ zugewiesen.

Jede dieser Operationen darf nur ausgeführt werden, wenn sich dadurch der Tabellentyp nicht ändert. Die Metadatenbank muß so konzipiert und kontrolliert werden, daß nur Veränderungen an Metadaten erlaubt sind, die den vorgenannten Operationen entsprechen. Ausgenommen hiervon ist die Schaffung neuer Tabellentypen.

Diese weitreichenden Einschränkungen sind notwendig, um die effiziente Verwaltung temporaler Daten in zeitveränderlichen Strukturen zu ermöglichen, ohne das jede Strukturänderung die Schaffung neuer Tabellen nach sich zieht. Eine solche Vorgehensweise wäre flexibler, aber auch ungleich schwerer zu beherrschen. Die ohnehin sehr hohe Anzahl von Tabellen der ATKIS-Datenbank würde sich weiter vergrößern, Recherchen zu Objektgeschichten würden erschwert.

### **5.3 Zusammenfassung**

Die in diesem Kapitel aufgestellten Grundsätze führten bislang noch nicht zur Schaffung eines temporalen Metainformationssystems. Sie fanden aber Beachtung bei Konzeption und Implementation des in Abschnitt 6 vorgestellten Metainformationssystems DST, das auf einfache Art und Weise um temporale Attribute erweitert werden kann. Diese Aufgabe und die Umsetzung der gegebenen Anregungen müssen Zielsetzung weiterführender Bemühungen sein.



## 6 Datenstrukturtabellen

Eine effiziente Erstellung und Verwaltung der ATKIS-Daten macht es unerlässlich, Meta-informationen über die komplexe Struktur der ATKIS-Daten und ihre begleitenden Bedingungen, insbesondere den Inhalt des OK, in einer leicht zu verarbeitenden Form abzulegen. Zu diesem Zwecke wurden die im folgenden beschriebenen sogenannten Datenstrukturtabellen, kurz DST, geschaffen. Sie werden mit Hilfe des relationalen Datenbanksystems Informix verwaltet und sind in einer Struktur angelegt, die das einfache Recherchieren verschiedener Informationen bezüglich der ATKIS-Datenstruktur, aber auch bezüglich Modellierungs- und Objektbildungsvorschriften erlaubt.

### 6.1 Anforderungen an die Datenstrukturtabellen

Für die Schaffung universeller Prozeduren zur Bildung und Manipulation von Objekten, zur automatischen Erzeugung der Datenstruktur für Objekte und zur Überprüfung verschiedener Integritätsbedingungen der ATKIS-Daten wäre die einfache Verfügbarkeit der im folgenden aufgelisteten Informationen von großem Interesse.

- Informationen über alle im OK aufgeführten und alle für die TÜK 200 zusätzlich erforderlichen Objektarten, einschließlich der Art ihrer Modellierung und der Zugehörigkeit zu topographischen Netzen (z.B. Straßennetz)
- alle im OK enthaltenen Attribute, ihre Eigenschaften und ihre Zuordnung zu den in Abschnitt 7.2 definierten Klassen
- eine Auflistung aller erlaubten Werte bezüglich jeden Attributes inklusive der Wertbedeutungen bei verschlüsselten Attributen
- eine Auflistung aller erlaubten Referenzen zwischen Objektarten (z.B. ist\_Teil\_von-Relation bei Komplexobjekten, Über- und Unterführungsreferenz)
- Informationen über die Relevanz einzelner Attribute in verschiedenen Erfassungsstufen
- Restriktionen bei der Überlagerung von flächenhaften Objekten

Es besteht die Möglichkeit, diese Daten in Form einer einfachen linearen Liste analog zum OK abzulegen. Aufgrund der zahlreichen Redundanzen innerhalb einer solchen Tabelle, wie z.B. der häufig auftretenden Mehrfachbenutzung eines Attributes für mehrere Objektarten, empfiehlt sich die Ablage in einer relationalen Struktur. Unter Beachtung aller einzubringenden Informationen entstand das in Abbildung 16 dargestellte Entity/Relationship-Modell

(E/R-Modell). Die Erläuterung der einzelnen Komponenten ist den Tabellen 24 und 25 in Anhang A.4 zu entnehmen.

Die Grundlage dieser Struktur bildeten folgende Überlegungen. Jede Objektart beinhaltet i. allg. mehrere Attribute. Ein Attribut kann aber auch bei mehreren Objektarten auftreten. Dabei sollten Attributschlüssel (dreistellige Mnemonik), Attributname und andere Eigenschaften für alle Objektarten und auch in allen DLM übereinstimmen. Der OK ist in diesem und auch in einigen anderen Punkten leider nicht absolut konsequent. Wie mit Ausnahmen und Unstimmigkeiten verfahren wurde, ist im Abschnitt 6.3 beschrieben.

Des weiteren besitzt jedes Attribut einen bestimmten Wertebereich. Seine Beschaffenheit ergibt sich aus der Art des Attributes. Bei stetigen Attributen (siehe Abschnitt 2.3.1), in der DST unter *Art\_der\_Werte* mit 's' verschlüsselt, wird der tatsächliche numerische Wert abgelegt. Informationen über den Wertebereich sind hier nicht notwendig, da beliebige Werte zulässig sind. Kodierte Attribute ('a' → „Aufzählung“) enthalten einen vierstelligen numerischen Schlüssel. Die Liste aller möglichen Codes mit ihren Bedeutungen ist ebenso zu erfassen wie ihre Relevanz für verschiedene Objektarten, da nicht jeder Code für alle Objektarten mit dem entsprechenden Attribut zulässig ist. Diese beiden Sachverhalte sind in den Relationen *erlaubt\_Wert* und *hat\_Wert* abgelegt, welche sich beide auf die Liste *Wert* beziehen. Die letzte Art von Attributen beinhalten klassifizierte Werte ('k') wie z. B. Breitenklassen von Flüssen. Alle definierten Klassen sind in der *Wert*-Tabelle gespeichert. Hier ist nur die *erlaubt\_Wert*-Relation von Interesse, Unterschiede zwischen Objektarten existieren hier nicht.

Weitere Relationen sind die im OK definierten Referenz-Beziehungen und die erlaubten Flächenüberlagerungen, jeweils zwischen zwei Objektarten.

Die Konzeption der OK für die DLM 200 und 1000 als reine Selektion aus dem OK 25 empfiehlt die Erfassung aller OK in einer Struktur. Hierfür wird in den Tabellen *Objektart*, *Attribut* und *Wert* eine Spalte *dml* für die Kennzeichnung des DLM eingeführt.

Weitere erwähnenswerte Bestandteile sind die Attributklasse in *hat\_Attribut*, welche den in Abschnitt 7.2 definierten Klassen entspricht, die Kategorie, hinter welcher sich die Attributkategorie des OK verbirgt, und die Belegungsmodalitäten, diese geben Auskunft, ob das entsprechende Attribut in der aktuellen ATKIS-Erfassungstufe Berücksichtigung findet.

### 6.1.1 ATKIS-fremde Objektarten

In allen bisherigen Überlegungen wurden nur ATKIS-Objektarten berücksichtigt. Dem IfAG obliegt aber neben dem Aufbau der DLM 200 und 1000 auch die Pflege der Topographischen Übersichtskarte Deutschlands im Maßstab 1 : 200 000 (TÜK 200). Diese wurde bisher als analoges herkömmliches Kartenwerk geführt. Der konzeptionelle Aufbau des DLM 200

gestattet prinzipiell die Ableitung eines DKM mit den qualitativen und quantitativen Anforderungen der TÜK 200. In der geplanten ersten Ausbaustufe des DLM 200 sind jedoch einige entscheidende Informationen, sowohl auf Objekt- als auch auf attributiver Ebene, nicht vorhanden. Des weiteren gibt es Detailanforderungen an die TÜK 200, deren Verwirklichung im ATKIS so nicht vorgesehen ist.

Um die Herstellung der TÜK 200 dennoch sicherzustellen, wurde es notwendig, im Rahmen dieser Arbeit einige zusätzliche Objektarten, Attribute und Attributwerte einzuführen. Dabei wurde große Sorgfalt auf die Einhaltung der durch die OK vorgegebenen Konventionen, sowohl in begrifflicher als auch in inhaltlicher Sicht, gelegt. Das ermöglicht unter anderem das unproblematische Einfügen in die DST und die Verwendung aller ATKIS-Werkzeuge.

In den DST werden diese Objektarten, Attribute und Attributwerte durch die Einführung der Spalte *quelle* in den Tabellen *objektart*, *hat\_attribut* und *hat\_wert* eindeutig als IfAG-interne Bestandteile gekennzeichnet.

Die Aufnahme dieser zusätzlichen Elemente hat keinerlei Auswirkungen auf den Aufbau des DLM 200. Sie sind grundsätzlich nur für den internen Gebrauch im IfAG vorgesehen und werden in keinem Fall an Dritte weitergegeben. Mit dieser Prämisse ist der ATKIS-getreue Aufbau des DLM 200 trotz der Ergänzungen sichergestellt.

Eine Aufstellung aller IfAG-internen Objektarten, Attribute und Attributwerte ist in Anhang A.1 zu finden.

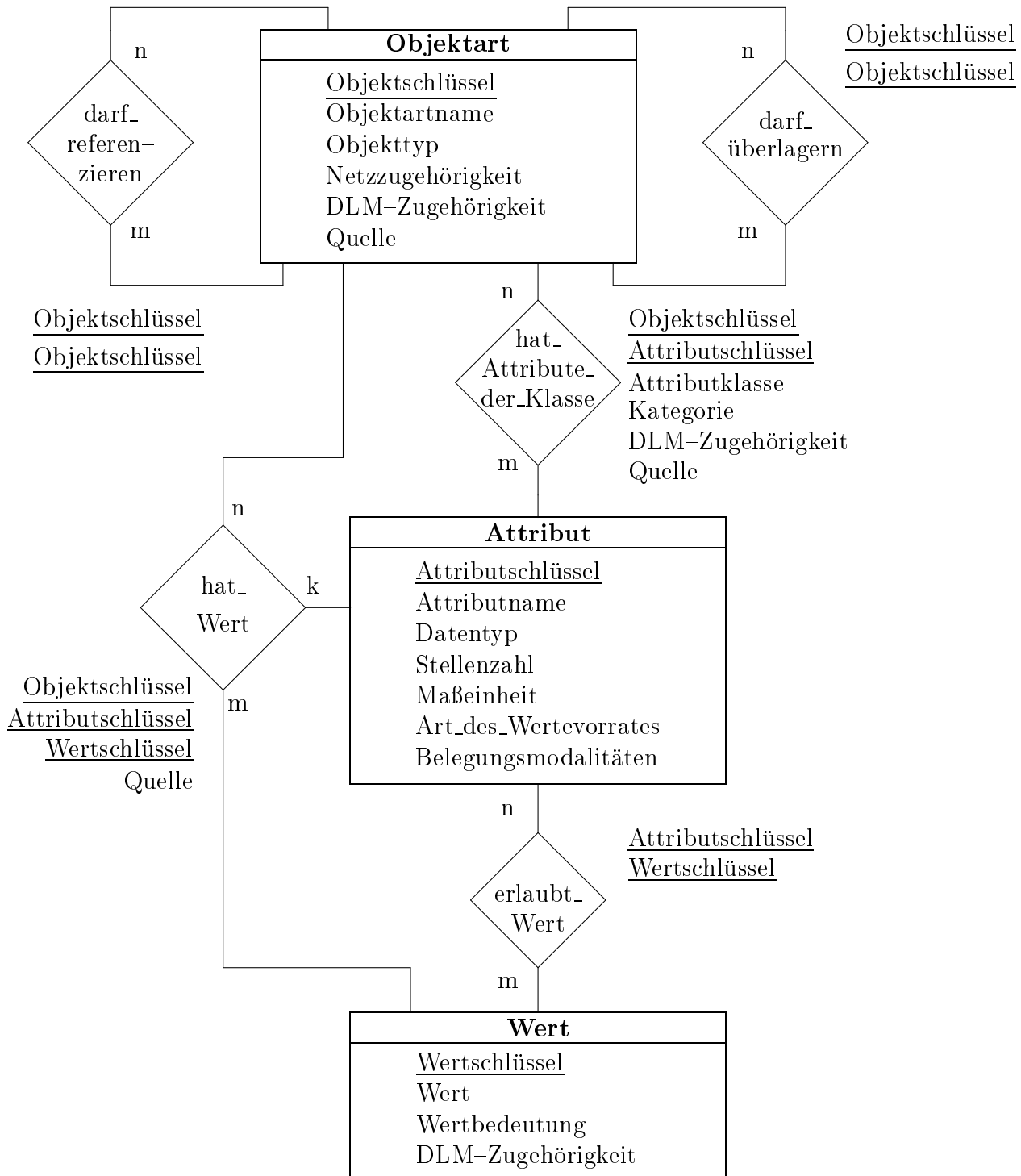


Abbildung 16: Entity/Relationship-Modell der Datenstrukturtabellen

## 6.2 Datenmodell

Ausgehend von dem in Abbildung 16 dargestellten Modell ergibt sich zunächst folgendes Relationenschema:

*Objektart* (Objektschlüssel, Objektartname, Objekttyp, Netzzugehörigkeit, DLM-Zugehörigkeit, Quelle)  
*darf\_referenzieren* (Objektschlüssel, Objektschlüssel)  
*darf\_überlagern* (Objektschlüssel, Objektschlüssel)  
*hat\_Attribute\_der\_Klasse* (Objektschlüssel, Attributschlüssel, Attributklasse, Kategorie, Quelle)  
*Attribut* (Attributschlüssel, Attributname, Datentyp, Stellenzahl, Maßeinheit, Art\_des\_Wertevorrates, Belegungsmodalitäten, DLM-Zugehörigkeit)  
*erlaubt\_Wert* (Attributschlüssel, Wertschlüssel)  
*hat\_Wert* (Objektschlüssel, Attributschlüssel, Wertschlüssel, Quelle)  
*Wert* (Wertschlüssel, Wert, Wertbedeutung, DLM-Zugehörigkeit)

Vor der Umsetzung dieses Schemas in eine reale Datenstruktur empfehlen sich die nachstehend erläuterten Änderungen, welche schließlich zu folgendem Schema führen:

*objektart* (oar, name, typ25, typ200, typ1000, netz, dlm, ueberlagern, quelle)  
*ist\_teil\_von* (oar1, oar2)  
*liegt\_ueber* (oar1, oar2)  
*hat\_attribut* (oar, attribut, klasse, kategorie, dlm, quelle)  
*attribut* (attribut, name, typ, stellenzahl, me, me\_1000, art, belegung)  
*wert* (attribut, wert, dlm, bedeutung)  
*hat\_wert* (oar, attribut, wert, dlm, quelle)

Es ist sinnvoll, die in der Relation *darf\_referenzieren* enthaltenen Beziehungen Komplexobjekt  $\longleftrightarrow$  Objekt und überführendes Objekt  $\longleftrightarrow$  unterführendes Objekt zu unterscheiden und in eigenen Tabellen zu verwalten. Recherchen im OK ergaben, daß jede Objektart, die eine andere überlagern darf, jede beliebige Objektart überlagern darf. Dies macht die separate Führung der Relation *darf\_überlagern* überflüssig, es genügt, ein Flag bei *Objektart* einzuführen, welches die notwendige Information beinhaltet. Bei *Objektart* muß außerdem der Objekttyp für die verschiedenen DLM unterschieden werden, dies führte zu den Einträgen *typ25*, *typ200* und *typ1000*.

In der Tabelle *Attribut* entstand zusätzlich *me\_1000*. Dieser Eintrag beinhaltet Maßeinheiten für das DLM 1000, sofern diese nicht mit denen der anderen DLM übereinstimmen.

Des weiteren hat sich bei genauerem Studium des OK herausgestellt, daß Werte nur sehr selten mehreren Attributen zugeordnet werden können, die *hat\_Wert*-Relation somit

als n-zu-1-Beziehung modelliert werden kann. Die dabei auftretenden geringfügigen Redundanzen stehen in keinem Verhältniss zu dem Aufwand, der mit der Führung eines separaten Wertschlüssels betrieben werden müßte, insbesondere, da die Datenstrukturtabellen nach einmaliger Erstellung kaum Änderungen unterworfen sein werden. Die eindeutige Identifizierung erfolgt über einen Schlüssel, welcher sich aus *Objektschlüssel*, *Attributschlüssel* und *DLM-Zugehörigkeit* zusammensetzt.

Außerdem sind aus technologischen Gründen mehrere Namensänderungen gegenüber dem ersten Schema notwendig; das verwendete RDBMS Informix akzeptiert zum Beispiel nur Bezeichner mit einer Länge bis 12 Zeichen. Unter Beachtung dieser Aspekte ergibt sich das oben aufgeführte, entgültige Schema:

Alle in diesem Schema auftretenden Komponenten sind in den Tabellen 24 und 25 in Anhang A.4 mit ihren zulässigen Werten aufgeführt. Zu einigen sind jedoch noch nähere Erläuterungen notwendig.

In den mehrfach auftretenden Spalten *dml* steht der Eintrag *a* für das DLM 25, *b* bedeutet die Zugehörigkeit zu den DLM 25 und 200, *c* steht für die DLM 25, 200 und 1000. Normalerweise sollten diese Codes alle auftretenden Fälle abdecken. Wie in den Abschnitten 6.3.1 bis 6.3.5 beschrieben, wird jedoch durch die aktuellen Ausgaben der OK in vielen Fällen die Selektionsbedingung (siehe Abschnitt 2.3.1) verletzt. So sind zum Beispiel Attribute des OK 200 in manchen Fällen nicht im OK 25 vorhanden. Dies machte die nachträgliche Einführung weiterer Codes für *dml* notwendig. So steht *d* für Werte, die *nur* im DLM 200 gelten, *e* für Werte, die ausschließlich dem DLM 1000 vorbehalten sind, *f* für DLM 200 und 1000, und *g* für die DLM 25 und 1000. Die etwas unübersichtliche Reihenfolge der Vergabe dieser Codes ist auf die Entstehungsgeschichte der DST zurückzuführen. Als die Notwendigkeit einer Erweiterung der *dml*-Spalte erkannt wurde, waren die bis dato verwendeten Codes bereits in eine Reihe von Tools und Routinen eingefloßen, deren Änderung einen unnötigen Aufwand dargestellt hätte.

Der Eintrag *netz* bezieht sich zur Zeit nur auf die Gewässer-, Straßen- und Eisenbahnnetze. An dieser Stelle können später auch Zugehörigkeiten zu anderen, z. B. Flächennetzen, abgelegt werden.

Die bei *wert* eingetragenen Werte für Attribute mit klassifiziertem Wertebereich sind in der Regel nicht vollständig. Der OK verwendet an dieser Stelle Definitionen der Art

bis 6 m	(Klasse 6)
über 6 m bis 9 m	(Klasse 9)
über 9 m bis 12 m	(Klasse 12)
über 12 m bis 15 m	(Klasse 15)

usw. in Schritten von 3 m.

[ATKIS]

Nachfolgende Klassen wurden nur bis zu einer sinnvoll erscheinenden Klasse abgelegt.

Die hier beschriebenen DST besitzen die Versionsnummer 1.8 . Eine genaue Kennzeichnung der Version ist notwendig, da die DST später um die in Abschnitt 5 beschriebene Zeitkomponente erweitert werden soll, die in Anhang B aufgeführten Tools aber auf die hier vorgestellte Struktur der DST zugreifen.

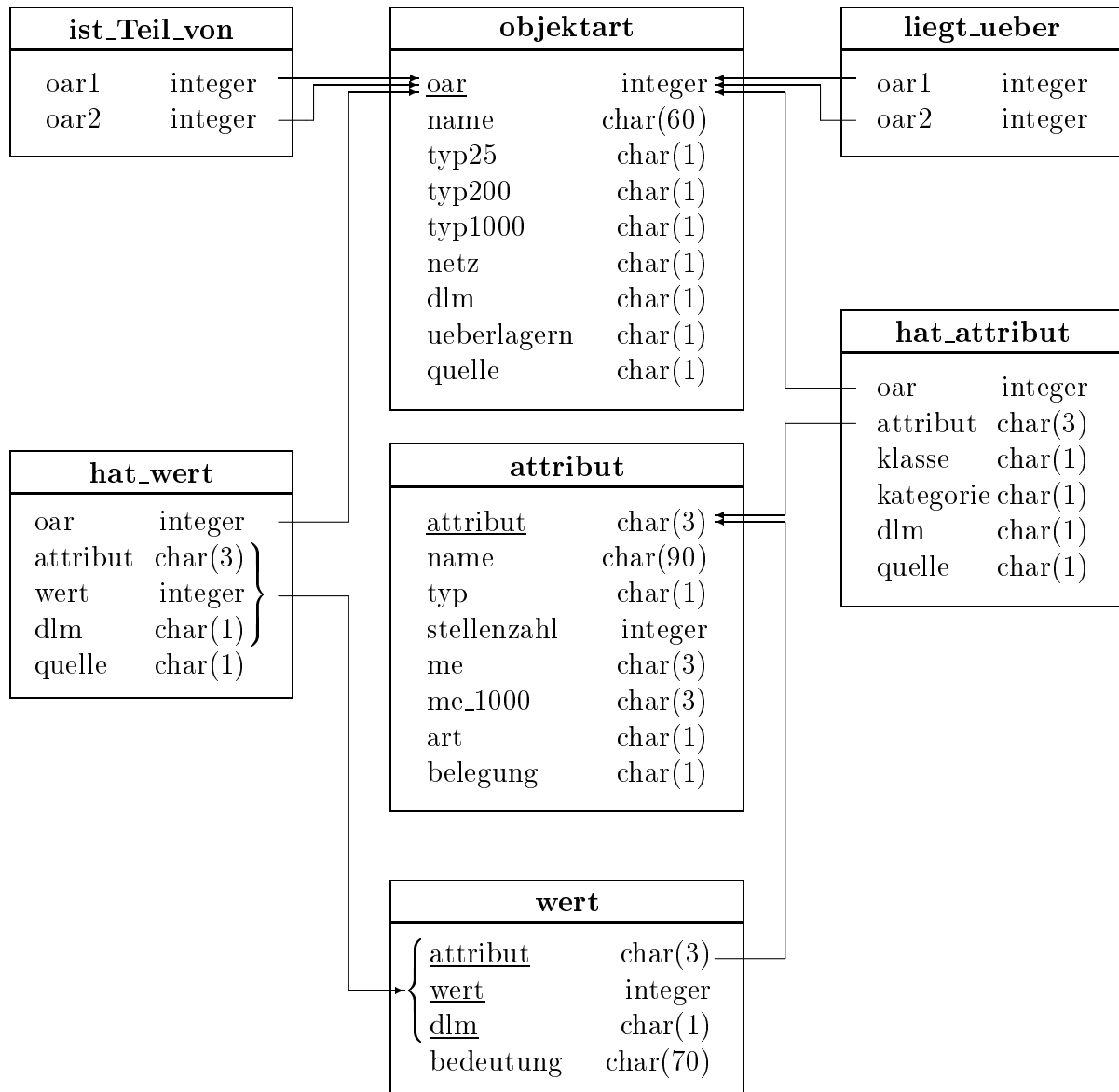


Abbildung 17: Die Tabellen der Datenstrukturtabellen und ihre Beziehungen (Version 1.8)

### 6.3 Realisierung

Die Umsetzung der in den letzten Abschnitten besprochenen Datenstrukturtabellen erfolgte durch die in Anhang A.3 aufgelisteten SQL-Anweisungen unter dem relationalen Daten-

banksystem Informix. Die durch diese Deklarationen entstandene Struktur ist in Abbildung 17 dargestellt. Dabei wurden alle Primär–Fremdschlüsselbeziehungen durch einen Pfeil dargestellt. Zu erwähnen sind hier nur der aus drei Komponenten bestehende Primärschlüssel in der Tabelle *wert* und die nur noch implizit über den Eintrag *attribut* bestehende Beziehung von *hat\_wert* und *attribut*.

Um die korrekte Erfassung der OK zu überprüfen, wurde die in Anhang B.1 dokumentierte AML–Routine `db_report.aml` entwickelt. Sie gibt den Inhalt der DST in einer OK–ähnlichen Form aus und ermöglicht so den Vergleich mit dem Original. Ein Beispiel hierfür ist in Abbildung 18 dargestellt (vergleiche dazu Abbildung 5).

Bei der Analyse der drei OK für die Füllung der DST kamen einige Unstimmigkeiten, insbesondere zwischen den OK, zutage. In den nachfolgenden Abschnitten werden alle aufgetretenen Fehlerklassen und die Art und Weise der Behandlung besprochen. Typische Beispiele werden jeweils in einer kleinen Tabelle dargestellt. Einige der Probleme sind auf den unterschiedlichen Bearbeitungsstand der OK zurückzuführen, bei den OK 200 und 1000 handelt es sich lediglich um Arbeitsversionen, andere entstanden zum Beispiel durch Änderungen bei Objektarten ohne Beachtung der Auswirkungen auf andere Objektarten. Diese Fehler konnten i. allg. in Absprache mit dem für die Entwicklung und Pflege der OK 200 und 1000 verantwortlichen Vertreter des IfAG bereinigt werden. Generell ist anzumerken, daß eine schnelle, unbürokratische Umarbeitung nur in den OK 200 und OK 1000 aufgrund ihres Status als Arbeitsversionen vorgenommen werden konnte.

Ein vollständige Liste aller gefundenen Unstimmigkeiten, die Art ihrer Bereinigung bzw. die Art der vorläufigen Abbildung des fehlerhaften Zustandes in den DST befindet sich in Anhang A.5.

### 6.3.1 Bezeichnungsfehler (1)

Objektart	OK	Objektartname
3531	25	Freileitung
	200	Freileitung
	1000	Kabelleitung
6100	25	Digitales Geländehöhenmodell
	200	Gelände
	1000	Digitales Geländehöhenmodell

Tabelle 2: Bezeichnungsfehler bei Objektarten

Bezeichnungsfehler kommen im Objekt– und im Attributbereich vor. Tabelle 2 zeigt



OK 200 c				
Objektart:	3102 Weg			
Objekttyp:	Linienförmig			
Quelle:	ATKIS			
BEF:	Befestigung			
Klasse: 3	Kategorie: 1	Art: a	Quelle: ATKIS	
Datentyp: i	Maßeinheit:			
1000	befestigt			
9997	Attribut trifft nicht zu			
9998	nach Quellenlage derzeit keine Zuweisung möglich			
FKT:	Funktion			
Klasse: 2	Kategorie: 1	Art: a	Quelle: ATKIS	
Datentyp: i	Maßeinheit:			
1701	Hauptwirtschaftsweg, Verbindungsweg (Fahrweg)			
1702	Wirtschaftsweg (Feld-, Waldweg)			
1703	Fußweg, Karren- und Ziehweg, Wattenweg			
1709	(Kletter-)Steig im Gebirge			
9998	nach Quellenlage derzeit keine Zuweisung möglich			
GN:	Geographischer Name			
Klasse: 1	Kategorie: 1	Art: s	Quelle: ATKIS	
Datentyp: c 50	Maßeinheit:			
KN:	Kurzbezeichnung			
Klasse: 1	Kategorie: 1	Art: s	Quelle: ATKIS	
Datentyp: c 50	Maßeinheit:			
ZN:	Zweitname			
Klasse: 1	Kategorie: 1	Art: s	Quelle: ATKIS	
Datentyp: c 50	Maßeinheit:			
ZUS:	Zustand			
Klasse: 3	Kategorie: 1	Art: a	Quelle: ATKIS	
Datentyp: i	Maßeinheit:			
1100	in Betrieb			
1200	außer Betrieb, stillgelegt			
1300	im Bau			
9998	nach Quellenlage derzeit keine Zuweisung möglich			
Referenzen				
Objektteil oben:				
3513	Tunnel			
3514	Brücke, Überführung, Unterführung			
Referenzen				
Objektteil unten:				
5112	Binnensee, Stausee, Teich			
3514	Brücke, Überführung, Unterführung			
3513	Tunnel			
5101	Strom, Fluß, Bach			
5103	Graben, Kanal (Wasserwirtschaft)			
5302	Talsperre, Wehr			

Abbildung 18: Ausgabeseite der AML-Routine `db_report.aml`

einige Beispiele für uneinheitliche Bezeichnungen für Objektartnamen. Diese Fehler sind im Hinblick auf die Selektionsregel der OK (siehe Abschnitt 2.3.1) nicht zu vertreten. Für die DST wurde jeweils nur ein einheitlicher Objektartname ausgewählt. Eine entsprechende Korrektur in den Arbeitsversionen der OK ist vorgesehen.

OK	Objektart	Attributkürzel	Attributname
25	7201	FLB	Fläche(berechnet)
25	7211	FLA	Fläche(amtlich)
200	7101	FLA	Fläche(amtlich)
200	7201	FLB	Fläche(berechnet)
200	7211	FLB	Fläche(amtlich)
1000	7211	FLA	Fläche(amtlich)

Tabelle 3: Beispiele für die Verwendung von FLA und FLB

Bei Attributen treten zwei Arten von Bezeichnungsfehlern auf. Zum einen sind Kurzbezeichnungen für Attribute uneinheitlich vergeben worden. Ein Beispiel hierfür ist die aus Tabelle 3 ersichtliche unterschiedliche Verwendung von FLA und FLB in Kombination mit Fläche(amtlich) und Fläche(berechnet). Fehler dieser Art wurden inzwischen in Absprache mit dem Verantwortlichen im IfAG behoben.

Zum anderen wurden für ein Kürzel mehrere Attributbezeichnungen vergeben. Siehe dazu Tab. 4. In diesen Fällen wurde für alle Attribute ein Vertreter ausgewählt, z.B. Produkt für PRO. Da dieser Fall nur recht selten auftritt, wurde auf die doch recht aufwendige Anpassung der DST verzichtet, zumal damit zu rechnen ist, daß in zukünftigen Ausgaben der OK diese Mehrdeutigkeiten beseitigt werden.

### 6.3.2 Zusammenfassungsfehler (2)

Die bei den DLM 200 und 1000 übliche Zusammenfassung von Attributwerten des DLM 25 erfolgte in einigen Fällen fehlerhaft oder zumindest im Widerspruch zur Selektionsbedingung (siehe Abschn. 2.3.1, Seite 10). Einige der zwischen DLM 200 und 1000 differenten Zusammenfassungen werden durch die unterschiedliche Aktualität der OK bedingt und in der nächsten Überarbeitung behoben. Diese wurden vorübergehend unter Verwendung verschiedener *dml*-Einträge in der Tabelle *hat\_attribut* abgelegt. Dadurch wurde auf einfache Art und Weise eine Möglichkeit gefunden, diese im Sinne der Konzeption des OK fehlerhaften Attributwerte abzulegen und zu verwenden. Es sind jedoch Konstellationen denkbar, unter denen dieser Mechanismus nicht ausreichend ist. Würde sich also das zuständige Gremium

OK	Objektart	Attributkürzel	Attributname
25,200	2121,2301	PRO	Produkt (Abbaugut)
25,200	2125	PRO	Produkt (Lagergut)
25	2130	PRO	Produkt (Herstellungsgut)
25, 200	2313	PRO	Produkt (Speicherinhalt)
25	2314	PRO	Produkt
25	2315,2316,2333,2334	PRO	Produkt (Baumaterial)
25,200	3532	PRO	Produkt, Transportgut
25	7311	SZE	Schutzzone allgem. oder gegen qualitative Beeinträchtigung
200	7311	SZE	Schutzzone

Tabelle 4: Beispiele für die mehrdeutige Verwendung von Attributkürzeln

der AdV gegen die vollständige Durchsetzung ihrer ursprünglichen Konzeption entscheiden, so wäre über eine nachträgliche strukturelle Änderung der DST nachzudenken.

ATT	OAR	OK 25		OK 200	
PRO	2301	1000	Erde	1000	Erde, Abraum, Geröll
		1005	nicht vorhanden		
		2300	nicht vorhanden		
	2302	1000	Erde	1000	Erde, Abraum, Geröll
		1005	Abraum		
		2300	Geröll		
OFM	4120	2200	Steine, Schotter	2200	Steine, Schotter, Geröll
		2300	Geröll		
	6207	2200	Steine, Schotter	2200	Steine, Schotter, Geröll
		2300	nicht vorhanden		
	6205	2200	nicht vorhanden	2200	Steine, Schotter
		2300	nicht vorhanden		

Tabelle 5: Beispiele für Zusammenfassungsfehler bei Attributwerten

### 6.3.3 Inhaltliche Fehler (3)

Eine große Anzahl dieser Fehler ist auf die unerlaubte Verwendung des Standardattributwertes *sonstiges* (9999) für andere Aussagen zurückzuführen. Solch eine unerlaubte Verwendung ist nicht als gravierender Fehler anzusehen, wohl aber als Verstoß gegen die Prinzipien des OK. Diese anhand einiger Beispiele in Tabelle 6 aufgezeigten Fälle bedürfen noch einer Klärung. Vorerst wurden sie fehlerhaft in die DST übernommen. Hieraus ergeben sich keinerlei Einschränkungen bezüglich des Aufbaus eines korrekten DLM.

Die letzten beiden Fallbeispiele in Tabelle 6 stehen stellvertretend für eine Reihe von Fehlern, welche aus einer ungenügenden Synchronisation der Schreibweisen und Bezeichnungen zwischen den einzelnen OK resultieren. Teilweise mag auch der unterschiedliche Bearbeitungsstand der OK dafür verantwortlich zeichnen.

Objektart	ATT	OK	Attributwert
2121	PRO	25	9999 sonstiges
		200	9999 Lehm, Lava, Mergel, Quarz, Schiefer, Kalkstein, Granit, sonstiges
		1000	9999 sonstiges
2125	PRO	25	9999 sonstige
		200	9999 Futtermittel, sonstige
7101	ADM	25	4007 ursprünglich gemeindefreies Gebiet
		200	4007 ursprünglich gemeindefreies Gebiet
		1000	4007 selbstständige Gemeinde
5101	HYD	25	2000 nicht ständig wasserführend
		200	2000 nicht ständig wasserführend
		1000	2000 nicht ständig periodisch wasserführend

Tabelle 6: Beispiele für inhaltliche Fehler bei Attributwerten

### 6.3.4 Unvollständigkeiten (4)

Der Vergleich der drei OK bringt eine Reihe der in Tabelle 7 mit Beispielen belegten Unvollständigkeiten auf Attribut- und Attributwertebene zutage. In den meisten Fällen sind Attribute bzw. Werte aus den OK 200 oder 1000 im OK 25 oder 200 nicht enthalten. Das stellt einen klaren Verstoß gegen das Selektionsprinzip dar (siehe 2.3.1, Seite 10). Diese Fehlerkategorie ist auf den unterschiedlichen Bearbeitungsstand der OK zurückzuführen und wird mit der nächsten offiziellen Ausgabe der OK behoben sein..

Die Struktur der DST läßt über die Verwendung der Codes für die *dIm*-Spalten auch die Ablage dieser fehlerhaften Informationen zu.

Objektart	Attribut	Wert	OK 25	OK 200	OK 1000
2315	GFK	1303	fehlt	vorhanden	
3102	FKT	9999	vorhanden	fehlt	vorhanden
6211	OFL	1700	fehlt	fehlt	vorhanden
3106	WDM		fehlt	vorhanden	vorhanden
3106	BDI		fehlt	vorhanden	
3532	DMS		vorhanden	fehlt	vorhanden
5101	PRB		fehlt	vorhanden	vorhanden

Tabelle 7: Beispiele für fehlende Attribute und Attributwerte

### 6.3.5 Sonstige Fehler(5)

Dieser Abschnitt beschreibt alle Fehlerarten, die nicht in eine der anderen Kategorien einzuordnen sind. Es handelt sich hierbei um unbedeutende oder nur einzeln auftretende Unstimmigkeiten.

So sind in der aktuellen Fassung der OK viele Überführungsreferenzen nur in einer Richtung angegeben. Nach Auskunft der AdV handelt es sich hierbei um ein rein editorielles Problem, das spätestens mit der nächsten offiziellen Ausgabe behoben sein wird. In den DST wurden alle Referenzen korrekt abgelegt.

Eine weiterer Mangel ist der Wechsel des Attributtyps bei BRF von stetig (tatsächliche Werte) im OK 25 zu klassifiziert in den OK 200 und 1000. Das stellt an sich keine Diskrepanz zu den Anforderungen an die OK dar, kann jedoch in dieser Form nicht in den DST gespeichert werden. Da dies eine einmalige Ausnahme unter den ohnehin schon selten auftretenden stetigen Attributen ist, wird an dieser Stelle der Einfachheit halber auf eine korrekte Dokumentation in den DST verzichtet.

Bei einigen Attributen wurde es versäumt, an entsprechender Stelle die zugelassenen Attributwerte aufzuführen. Ob für die betroffenen Objektarten alle im allgemeinen Index aufgeführten Attributwerte zulässig sind, ist noch zu klären. Vorerst wurde von dieser Sicht ausgegangen.

Kleine Unstimmigkeiten in Inhaltsverzeichnissen und Indizes stellen mit Sicherheit keine gravierenden Fehler dar, sollen aber hier, und damit im Unstimmigkeitenkatalog (siehe Anhang A.5), der Vollständigkeit wegen mit erwähnt werden.

## 7 DLM–Datenmodell

Durch die ATKIS–Dokumentation wird nur ein grobes Schema für die Modellierung und Speicherung der Daten des DLM vorgegeben. Diese in Abschnitt 2.4 beschriebenen, konzeptionellen Vorstellungen werden nun im Hinblick auf eine spezielle Implementation überdacht und spezifiziert. Dazu gehören auch spezielle Anforderungen, die sich durch die Verwendung des GIS ARC/INFO ergeben (siehe Abschnitt 3).

Eine Implementation des ATKIS–DLM am IfAG sollte folgende Aspekte verwirklichen:

- die redundanzfreie, eindeutige Abbildung der geometrischen Elemente (spatiale Daten) der Landschaftsobjekte,
- die korrekte Abbildung der Struktur der ATKIS–Objekte und –Komplexobjekte, ihrer Attribute und ihrer Beziehungen zueinander, einschließlich verschiedener Zusammenhänge wie topographische Netze und administrative Hierarchien,
- die Verwaltung sehr großer Datenmengen,
- eine Geschichtsverwaltung für Geometrien, Objektteile, Objekte und ihre Attribute sowie
- die Möglichkeit der späteren Erweiterung der Datenstruktur ohne Beeinträchtigung vorhandener Daten.

Unter Berücksichtigung dieser Gesichtspunkte wurde eine Datenstruktur einschließlich aller Randbedingungen entwickelt. Dabei wurde besondere Aufmerksamkeit auf die Diskussion der Redundanzfreiheit, der Offenheit der Struktur gegenüber Änderungen und der vollständigen Abbildung von ATKIS und speziellen Anforderungen des IfAG gerichtet.

Abschließend wurde das vorgeschlagene Modell anhand einer Probeinstallation bezüglich des Laufzeitverhaltens und der Handhabbarkeit bewertet. Diese Testinstallation verwirklicht aber nur einen Teil der vorgeschlagenen Struktur, die Abbildung von Geschichten wurde zunächst ausgelassen. Dieser Teil hätte den Rahmen dieser Arbeit gesprengt und muß Gegenstand weiterführender Untersuchungen sein.

### 7.1 Spatiale DLM–Daten

Laut ATKIS–Dokumentation werden in dem dort beschriebenen DLM die Lageeigenschaften der Landschaftsobjekte (kurz: Objekte), abgesehen von referentiellen Zusammenhängen, einzig und erschöpfend durch drei verschiedene Geometrietypen beschrieben: punkt-, linien- und flächenförmige. Als punktförmiger Geometrietyt ist die von ARC/INFO bereitgestellte

Featureklasse point ausreichend (siehe Abschnitt 3.2). Mit Hilfe der Klasse arc lassen sich alle linienförmigen Geometrien abbilden. Alle Arten von linearen Geometrien, die dem Wesen nach nicht aus Strecken bestehen, zum Beispiel Kreise, lassen sich unter Wahl einer angemessenen Stützpunktzahl mit Hilfe von arc's beliebig genau nachbilden. Flächengeometrien schließlich können in der Featureklasse poly abgelegt werden.

Diese im folgenden als Basiselemente bezeichneten drei Geometrielemente ermöglichen eine vollständige geometrische Beschreibung der DLM-Daten. Man könnte nun direkt an diese Basiselemente die Daten der Objektteilebene inklusive ihrer attributiven Informationen anfügen. Hierzu würde sich der ARC/INFO-Mechanismus zur Verwaltung aspatialer Daten anbieten, die Featureattributtabelle (FAT). Man müßte allerdings das Fehlen von Integritätsmechanismen in Kauf nehmen. Auch die in Abschnitt 7.2 diskutierten überflüssigen Attribute würden hier auftreten. Das größte Problem bei diesem Ansatz ist aber die nicht zu vermeidende Mehrfachspeicherung von identischen Geometrien. Solche Redundanzen verhindern effiziente und sichere Fortführungsverfahren und sind deshalb unakzeptabel.

In den folgenden Abschnitten sollen zwei Möglichkeiten beschrieben werden, die  $n : m$ -Beziehung zwischen Geometrieelementen und Objektteilen mit Mitteln von ARC/INFO redundanzfrei abzubilden.

### 7.1.1 Einfaches Geometriemodell

Bis zur Version 6.0 wurde von ARC/INFO die Bildung komplexerer Geometrietypen nicht unterstützt. Unter diesen Voraussetzungen war die Erschaffung eigener komplexer Geometrietypen, basierend auf den Basistypen, unerläßlich. Aber auch aus heutiger Sicht kann ein solcher Ansatz sinnvoll sein, da die zum Zeitpunkt der Erstellung dieser Arbeit aktuelle Version 7.04 noch eine Reihe von Problemen bei der Verwaltung der nun vorhandenen komplexen Geometrietypen Region und Route aufweist.

Es ist also sinnvoll, über den *einfaches Geometriemodell*<sup>1</sup> genannten Lösungsweg nachzudenken, der nur auf die geometrischen Basiselemente zurückgreift.

Die Basis dieses Ansatzes ist folgende Überlegung. Zur Abbildung der ATKIS-Objektteile wird die in Abbildung 19 dargestellte Struktur verwendet. Es werden nur die Featureklassen point, arc und poly verwendet. Um Redundanzen zu vermeiden, werden alle arc und poly in einem Cover abgelegt. Ein zweites Cover nimmt die Elemente der Featureklasse point<sup>2</sup> auf. Jedem geometrischen Basiselement wird ein eindeutiger Schlüssel zugeordnet, mit dem das Element referenziert werden kann. Die Abbildung der Objekt- und Komplexobjektebene

---

<sup>1</sup> Besser wäre *Geometriemodell mit einfachen Geometrietypen*. Der Begriff *einfaches Geometriemodell* ist aber unkomplizierter zu handhaben.

<sup>2</sup> Die Verwaltung im selben Cover wie arc und poly ist nicht möglich, da ein Cover unter ARC/INFO nicht Elemente der Featureklassen poly und point gemeinsam aufnehmen kann.

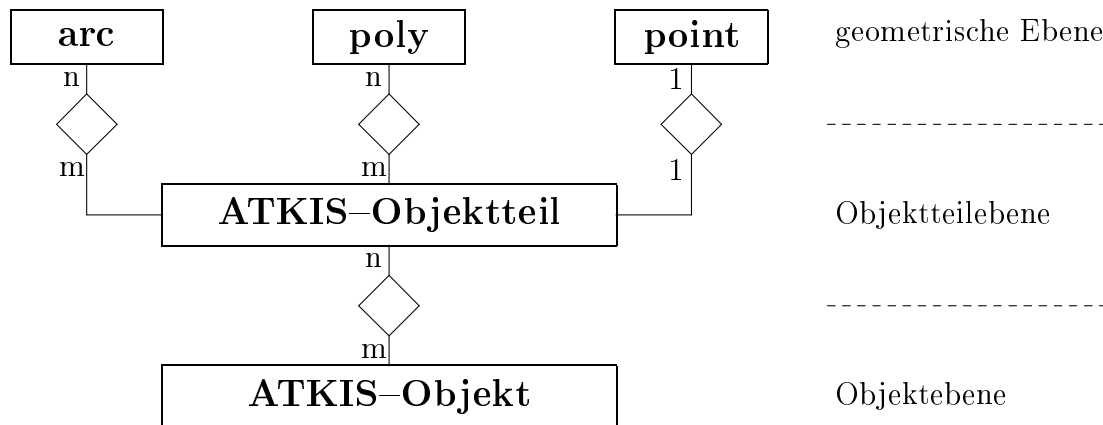


Abbildung 19: Einfaches Geometriemodell

ist von der geometrischen Repräsentation der Objektteile unabhängig und deshalb für einfaches und komplexes Geometriemodell identisch. Sie wird in Abschnitt 7.2 und folgenden beschrieben.

Das einfache Geometriemodell erfordert einen großen Programmieraufwand bei der Schaffung von Tools zur Fortführung. Funktionen von ARCEDIT zur Geometriemanipulation können nicht direkt verwendet werden. Bei jeder Änderungsoperation muß der Erhalt der Struktur von der zu schaffenden Software sichergestellt werden.

### 7.1.2 Komplexes Geometriemodell

Im Gegensatz zu dem im voraus beschriebenen Ansatz soll das komplexe Geometriemodell<sup>3</sup> vorhandene Funktionalität von ARC/INFO zur Bildung komplexer Geometrietypen nutzen. Wie bereits in den Abschnitten 3.2.1 und 3.2.2 vorgestellt, handelt es sich bei Routen und Regionen um Geometrietypen ohne eigene Lageeigenschaften, es sind lediglich logische Zusammenfassungen von vorhandenen `arc` bzw. `poly`. Sie sind damit den ATKIS-Linien- und Flächenobjektteilen konzeptionell sehr ähnlich<sup>4</sup>.

Die Struktur von Routen und Regionen erlaubt die redundanzfreie Verwaltung aller linien- und flächenhaften Geometrien. Dafür müssen aber alle komplexen und Basiselemente in einem Cover abgelegt sein<sup>5</sup>. Inwiefern das zu Laufzeitproblemen führen kann, ist noch zu untersuchen.

<sup>3</sup> Besser wäre *Geometriemodell mit komplexen Geometrietypen*. Siehe auch Fußnote 1.

<sup>4</sup> Wir werden bei diesen Überlegungen nur über linien- und flächenhafte Geometrietypen sprechen. Die Problematik der punktförmigen Objektteile kann aus dieser Diskussion ausgeschlossen werden, da ihre Abbildung unproblematisch wie beim einfachen Geometriemodell durch die Featureklasse `point` erfolgen kann. Ein Punktobjektteil wird immer nur durch einen einzigen Geometriepunkt repräsentiert, was die Bildung komplexer Strukturen überflüssig macht.

<sup>5</sup> Einzige Ausnahme können eventuell die Geometrien der Objektart Höhenlinie (OAR 6102) sein, diese ist als einzige geometrisch unabhängig von anderen Objektarten.



Für das komplexe Geometriemodell ergibt sich die in Abbildung 20 dargestellte Struktur.

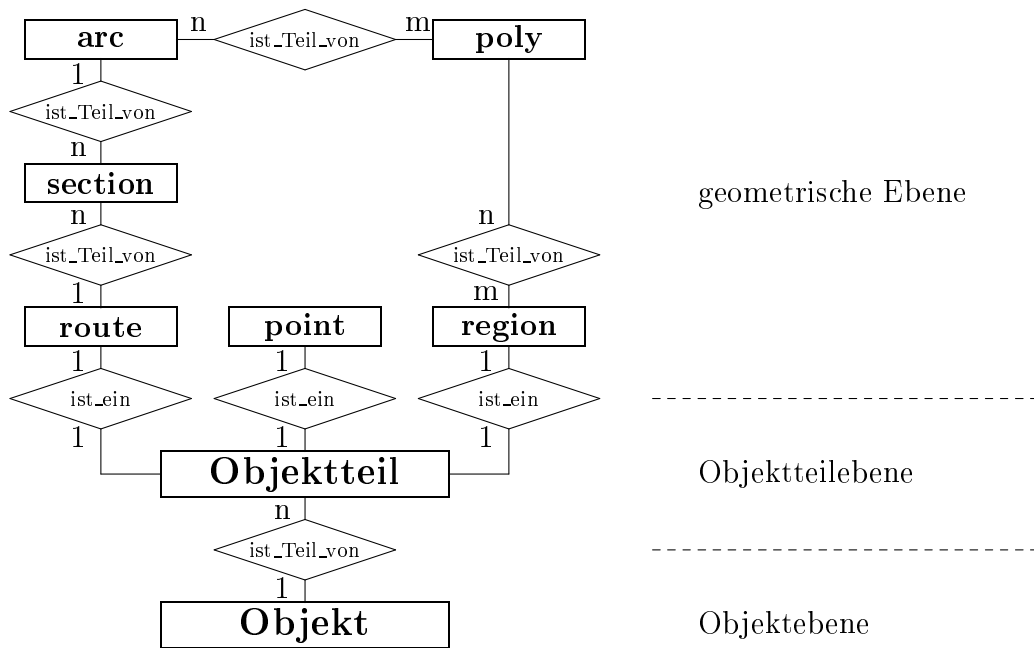


Abbildung 20: Komplexes Geometriemodell

Im Gegensatz zum einfachen Geometriemodell können hier ARCEDIT-Funktionen zur Geometriemanipulation verwendet werden, solange das Niveau der komplexen Geometrielemente nicht verlassen wird. Lediglich das Löschen von Geometrien und die Kontrolle über die Geschlossenheit linienförmiger Objektteile bedarf noch der Kontrolle seitens der Fortführungssoftware.

Bei den bisherigen Betrachtungen wurde noch nicht geklärt, wieviele verschiedene Regionen- bzw. Routenklassen für die Implementation des DLM benötigt werden. Es wäre denkbar, für jede Objektart eine eigene Klasse zu schaffen. Durch die noch in der Entwicklung begriffene Funktionalität der Routen- und Regionenverwaltung könnte dies jedoch zu einer unnötigen Komplizierung des Zugriffs auf Objektteile führen. Die Speicherung aller Objektteile in jeweils einer einzigen Routen- bzw. Regionenklasse ist zwar bei Vernachlässigung des attributiven Teils realisierbar, kann aber zu Laufzeitproblemen führen. Die Erfahrung zeigt, daß speziell der Zugriff auf Routenklassen mit einer großen Anzahl von Elementen unverhältnismäßig langsam ist. Bis solche grundlegenden Probleme von ARC/INFO behoben sind, wird man einen Kompromiß zwischen den beiden Möglichkeiten finden müssen, zum Beispiel indem jeweils eine Routen- oder Regionenklasse für eine Gruppe von Objektarten etabliert wird.

## 7.2 Attributive DLM–Daten

Die OK definieren eine Vielzahl verschiedener Attribute für jede Objektart. Diese Attribute können der Objektteil-, der Objekt- oder der Komplexobjektebene zugeordnet sein und können darüber hinaus sehr verschiedene Eigenschaften haben. In diesem Abschnitt soll ein Modell entworfen werden, mit dem alle Attribute effizient unter Vermeidung von Redundanzen verwaltet werden können.

Dies sollte unter Verwendung eines kommerziellen RDBMS erfolgen. Das Fehlen grundlegender Integritätskonzepte disqualifiziert das ARC/INFO-interne RDBMS INFO für diese Aufgabe (siehe dazu auch Abschnitt 3.1). Der Database Integrator macht zudem die Anbindung eines externen RDBMS unproblematisch.

Wie bereits in 2.4 erläutert, unterscheiden wir zunächst zwei Klassen von Attributen, die Objektteil- und die Objektattribute (OT- und OJ-Attribute). Um Redundanzen bei der Speicherung zu vermeiden, ist es notwendig, die Klasse der Objektattribute weiter zu differenzieren:

**Objektabhängige Attribute:** Eventuelle Übereinstimmungen dieses Attributes bei mehreren Objekten basieren nicht auf semantischen Abhängigkeiten und sollen nicht als solche verwaltet werden.

**Objektübergreifende Attribute:** Attribute, die aufgrund semantischer Abhängigkeiten für mehrere Objekte identisch sind.

Die Objektattribute lassen sich auch nach einem anderen Kriterium unterscheiden. Durch die Dreiteilung des DLM werden viele attributive Informationen mehrfach erfaßt und gespeichert. Für die Objektteilebene hat dies keine große Bedeutung, da aufgrund der von DLM zu DLM unterschiedlichen Modellierungsregeln kaum echte Beziehungen hergestellt werden können. Anders liegt der Fall bei Objekten. Insbesondere geographische Namen sind unabhängig vom DLM immer gleich, ebenso bestimmte Attribute wie die überörtliche Verkehrsbedeutung (BDU). Das führt zu folgender Klassifizierung:

**DLM-abhängige Attribute:** Bei der Abbildung eines realen Landschaftsobjektes in verschiedenen DLM kann dieses Attribut unterschiedlich belegt sein.

**DLM-übergreifende Attribute:** Bei der Abbildung eines realen Landschaftsobjektes ist die Belegung dieses Attributes unabhängig vom DLM immer gleich.

Aus der Kombination der genannten Differenzierungen ergeben sich die in Tabelle 8 aufgeführten Attributklassen. Jedes ATKIS-Attribut läßt sich in eine dieser Klassen einordnen.

Attributklasse	Ebene	abhängig von		Elemente
		Objekt	DLM	
OT-Attribute	OT	–	–	758
OJ- und DLM-abhängige	OJ	ja	ja	14
OJ-abhängige	OJ	ja	nein	488
DLM-abhängige	OJ	nein	ja	0
OJ- und DLM-übergreifende	OJ	nein	nein	62

Tabelle 8: Attributklassen und die Anzahl ihrer Elemente einschließlich IfAG-interner Attribute<sup>7</sup>

Wie aus Tabelle 8 ersichtlich, gibt es keine Attribute, die DLM-abhängig, aber objektübergreifend sind. Für die verbliebenen Klassen ergeben sich die in Abbildung 21 dargestellten Beziehungen.

In Abschnitt 7.4.1 werden wir noch zeigen, daß sich Redundanzen bei den DLM- und objektübergreifenden Attributen durch die Schaffung neuer Komplexobjektarten günstiger vermeiden lassen. Diese Attribute können dann den DLM-übergreifenden und objektabhängigen Attributen zugeordnet werden.

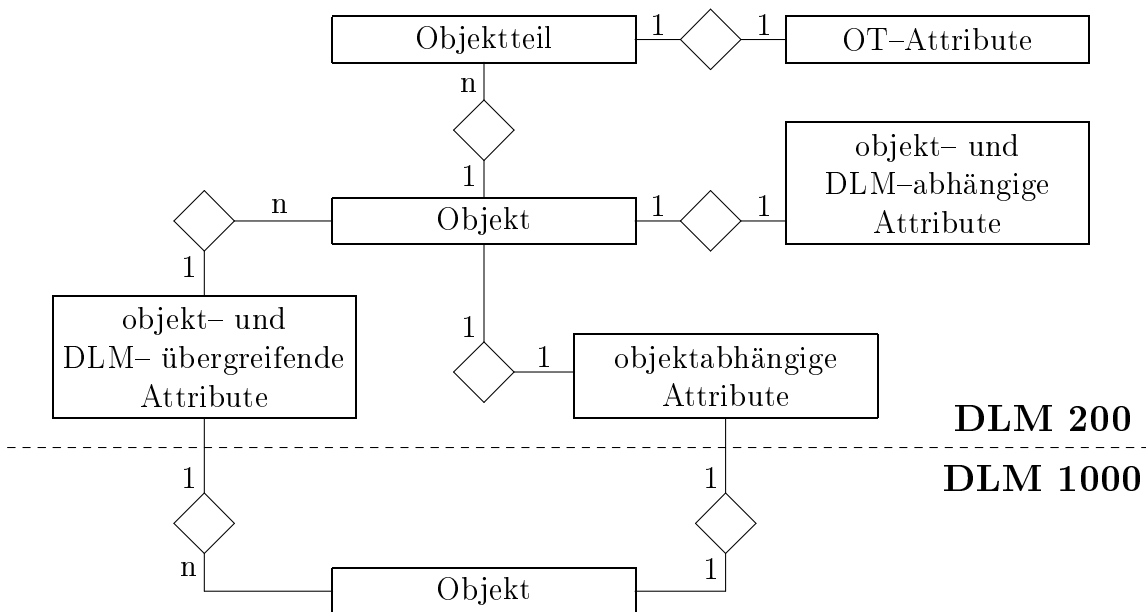


Abbildung 21: Die Attributklassen und ihre Beziehungen

<sup>7</sup> Bei der Anzahl der Elemente wurden Attribute, die bei mehreren Objektarten auftreten, mehrfach gezählt.

Die Zuordnung der ATKIS–Objektattribute zu den Klassen ist in Tabelle 16 in Anhang A.2 dokumentiert. Bei allen nicht in dieser Tabelle aufgeführten Attributen handelt es sich um Objektteilattribute.

Die drei verbliebenen Klassen (OT–Attribute, DLM– und objektabhängige Attribute, objektabhängige Attribute) müssen getrennt verwaltet werden. Dies kann für alle Objektarten gemeinsam oder getrennt nach Objektarten erfolgen. Die erste Variante ist unpraktikabel. Die Anzahl der tatsächlich benötigten Attribute für eine konkrete Objektart (durchschnittlich 5-10) ist gering gegenüber der Gesamtzahl aller Attribute (ca. 130). Eine große Menge unbenutzter Spalten wäre die Folge.

Das zweite Verfahren (getrennte Attributverwaltung für jede Objektart) bringt durch die hohe Anzahl der benötigten Tabellen (ca 360) eine Reihe von Schwierigkeiten mit sich. Die meisten Probleme lassen sich aber durch die Verwendung der DST (siehe Abschnitt 6) als Grundlage für alle Operationen entschärfen. Für viele Anwendungen einschließlich Erfassung und Fortführung kann die Speicherung der Attributdaten transparent bleiben, die Informationen für die Zuordnung zur richtigen Tabelle und zur Struktur dieser können aus den DST gewonnen werden.

Die vorgeschlagene Klassifizierung kann nicht alle Redundanzen verhindern, die wichtigsten Problemfälle aber, die redundanzfreie Speicherung der hervorgehobenen und der objektübergreifenden Attribute, sind mit der Umsetzung des vorgeschlagenen Klassenmodells vollständig gelöst.

### 7.3 Objekte und Objektteile

Die physische Anordnung der Objektteilebene ist abhängig von der Wahl des Geometriemodells. Das einfache Modell erfordert, wie in Abbildung 19 dargestellt, eine von der Geometrie physisch getrennte Speicherung der Objektteile. Das komplexe Geometriemodell dagegen erlaubt aufgrund der 1 : 1–Beziehung zwischen Objektteilen und komplexen Geometrietypen die gemeinsame Speicherung in den FAT. Die Speicherung der OT–Attribute sollten jedoch, wie bereits in Abschnitt 7.2 begründet, außerhalb von ARC/INFO erfolgen. Der nachfolgende Entwurf eines ATKIS–Datenmodells geht von der Nutzung des komplexen Geometriemodells aus. Alle Bemerkungen lassen sich aber leicht auf das einfache Geometriemodell übertragen.

Analog zu den Attributen sollen Objektteile und Objekte nach Objektarten getrennt verwaltet und gespeichert werden. Das heißt für jede Objektart wird eine Struktur wie in Abbildung 21 benötigt. Diese Strukturen unterscheiden sich nur in der Menge und Art der Attribute und in den Namen der Relationen, die Spalten für die Objekt– und Objektteilverwaltung sind identisch.

Unter Berücksichtigung aller Beziehungen ergibt sich aus Abbildung 21 folgende Datenstruktur:

*Objekt* (objektkey, Herkunft, Att\_OJ<sub>1</sub>, . . . , Att\_OJ<sub>n</sub>)

*Objektteil* (otkey, objektkey, Att\_OT<sub>1</sub>, . . . , Att\_OT<sub>m</sub>)

*übergreifende\_Att* (objektkey\_200, objektkey\_1000, Att\_übergr<sub>1</sub>, . . . , Att\_übergr<sub>k</sub>)

Diese Struktur ist vollständig normalisiert und beinhaltet keine Redundanzen.

Innerhalb einer Objektart erhalten alle Objekte und Objektteile einen eindeutigen Schlüssel (*objektkey* und *otkey*). Dabei besteht kein semantischer Zusammenhang zwischen dem *objektkey* eines Objektes und dem *otkey* seiner Objektteile. Je nach abgebildetem DLM referenziert entweder *objektkey\_200* oder *objektkey\_1000* zu *objektkey* der Relation *Objekt*. Att\_OJ<sub>1</sub>, . . . , Att\_OJ<sub>n</sub> ist die Menge der objekt- und DLM-abhängigen Attribute, Att\_OT<sub>1</sub>, . . . , Att\_OT<sub>m</sub> die Objektteilattribute und Att\_übergr<sub>1</sub>, . . . , Att\_übergr<sub>k</sub> die DLM-übergreifenden Attribute. Jede dieser Mengen kann auch leer sein.

Die Verbindung von Objektteilen und geometrischen Elementen erfolgt über den Schlüssel *otkey*. In Abbildung 20 ist zu sehen, daß ein Objektteil aus jeweils einem Geometrieelement von drei, in ARC/INFO getrennt verwalteten Geometrietyphen bestehen kann. Ein Objektteil kann jedoch *nicht* aus mehreren Geometrieelementen bestehen. ARC/INFO ist aber nicht in der Lage, die Eindeutigkeit des Objektteilschlüssels über mehrere Relationen (komplexe Geometrietyphen) hinweg sicherzustellen. Das Problem kann mit der im folgenden skizzierten Datenstruktur gelöst werden:

*route* (route\_ID)

*region* (region\_ID)

*point* (point\_ID)

*ist\_route* (route\_ID, otkey)

*ist\_region* (region\_ID, otkey)

*ist\_point* (point\_ID, otkey)

*objektteil* (otkey, Attribute. . . )

Diese Möglichkeit bedeutet aber eine Verdopplung der Tabellenzahl pro Objektart und damit einen erheblichen Mehraufwand. Es wurde deshalb eine Entscheidung zugunsten des ersten Modells getroffen. Das Problem der eindeutigen Schlüssel wurde durch eine Erweiterung des Objektteilschlüssels um den Geometrietyph gelöst. Die praktische Realisierung ist in Abschnitt 7.6 beschrieben.

Die beschriebene Struktur ist im Rahmen der in Abschnitt 5.1 eingeführten Begriffe ein

Tabellencluster mit den Tabellentypen *objekt*, *objektteil* und *übergreifende\_Att*. Der variable Teil besteht aus den Attributen der entsprechenden Klassen. Informationen über Tabellennamen, Attribute und Integritätsbedingungen jeder Objektart können dem Metainformationssystem DST entnommen werden (siehe Abschnitt 6). Dies sollte auch die Grundlage aller Operationen und Werkzeuge über der ATKIS-Datenstruktur sein, um die Flexibilität und Transparenz des ATKIS-Datenbestandes zu erhalten. Wie die entsprechenden Informationen aus den DST zu extrahieren sind, ist in 7.6 beschrieben.

An dieser Stelle soll bereits auf Abbildung 24 verwiesen werden, welche die gesamte ATKIS-Datenstruktur inklusive Objekt- und Komplexobjektmodellierung sowie Beziehungen zwischen Objektteilen und Objekten verschiedener Objektarten und verschiedener DLM darstellt.

### 7.3.1 Referenzen zwischen Objektteilen

Die ATKIS-Konzeption sieht über die Abbildung der Beziehungen von Objektteilen, Objekten und Komplexobjekten hinaus die Erfassung topologischer Zusammenhänge zwischen Objektteilen vor. Betrachtet werden Über- und Unterführungsreferenzen zwischen zwei Objektteilen, die sich in mindestens einem Punkt, einer Linie oder einer Fläche berühren oder überschneiden, aber räumlich in unterschiedlichen Höhen angeordnet sind. Im Sinne des Datenmodells sind Über- und Unterführungsreferenzen identisch, die Unterscheidung erfolgt nur aufgrund der unterschiedlichen Sichtweise vom unten bzw. oben angeordneten Objektteil.

Eine solche Referenz ist eine 1 : 1-Beziehung zwischen zwei Objektteilen der selben oder unterschiedlicher Objektarten. Welche Objektarten an Referenzen beteiligt sein können, ist im OK festgelegt. Danach sind über 60 Kombinationen erlaubt. Eine saubere Modellierung würde nun die Zuordnung jeder dieser Varianten zu einer eigenen Relation erfordern. Im Hinblick auf eine möglichst einfach zu handhabende Datenstruktur ist aber die Verwaltung aller Referenzen mittels einer Relation zumindest im Rahmen einer Prototypimplementierung vorteilhafter. Dazu müssen neben den zu referenzierenden Objektteilen auch die Objektarten abgelegt werden. Dieses Verfahren verhindert natürlich die Deklaration von referentiellen Integritätsregeln. Da dieses Problem prinzipiell auch bei der Verbindung von Objektteilen und Geometrien besteht, muß die Verwaltung der Objektteilschlüssel ohnehin von speziell dafür zu schaffenden Tools erfolgen, eine zusätzliche Kontrolle der Referenzen ist daher unproblematisch. Praktische Erfahrungen müssen zeigen, ob dieses Verfahren ausreichend ist, oder ob die ATKIS-Datenbank zu einem späteren Zeitpunkt um die entsprechenden Tabellen erweitert werden sollte.

## 7.4 Komplexobjekte

Komplexobjekte in ATKIS sind logische Zusammenfassungen von Objekten oder Komplexobjekten. Für jede Komplexobjektart ist festgelegt, welche Objekt- und Komplexobjektarten ihr angehören dürfen. Sie besitzen Attribute, die genau wie die Objektattribute klassifiziert werden. Die Zahl der bislang im OK aufgeführten Komplexobjektarten ist mit 3 sehr gering. Für diese Komplexobjektarten sind jeweils nicht mehr als zwei Objektarten als Bestandteil definiert. Auch die in Abschnitt 7.4.1 eingeführten IfAG-internen Komplexobjektarten erfüllen dieses Kriterium. Diese Eigenschaft ermöglicht es, die in Abbildung 22 dargestellte Struktur ohne die bei Referenzen aufgetretenen Probleme zu implementieren.

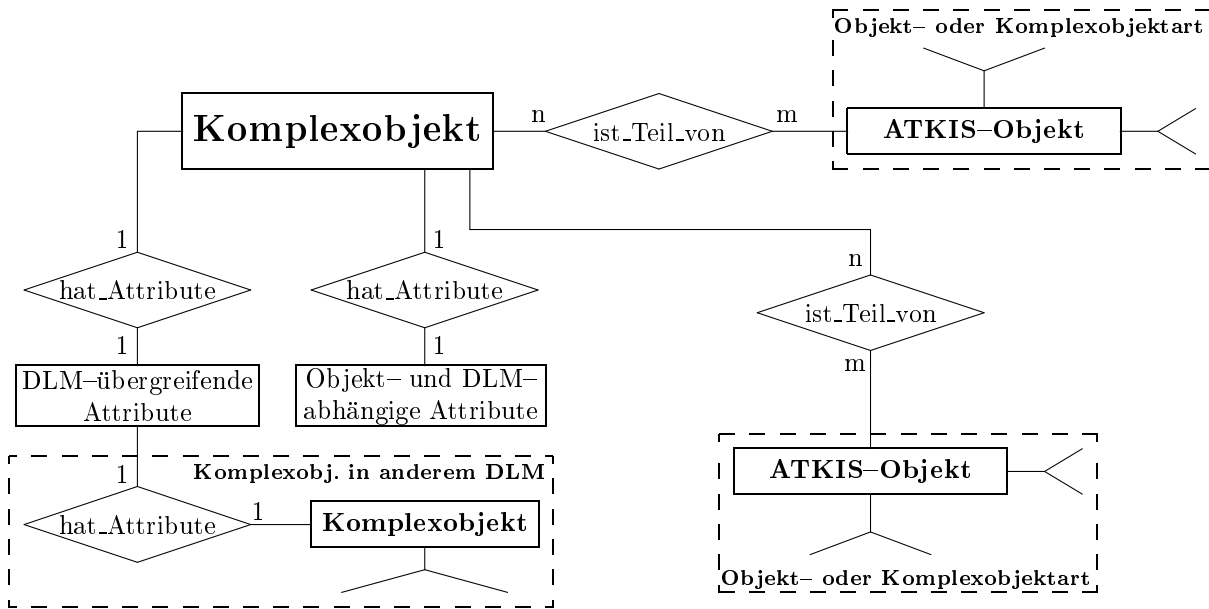


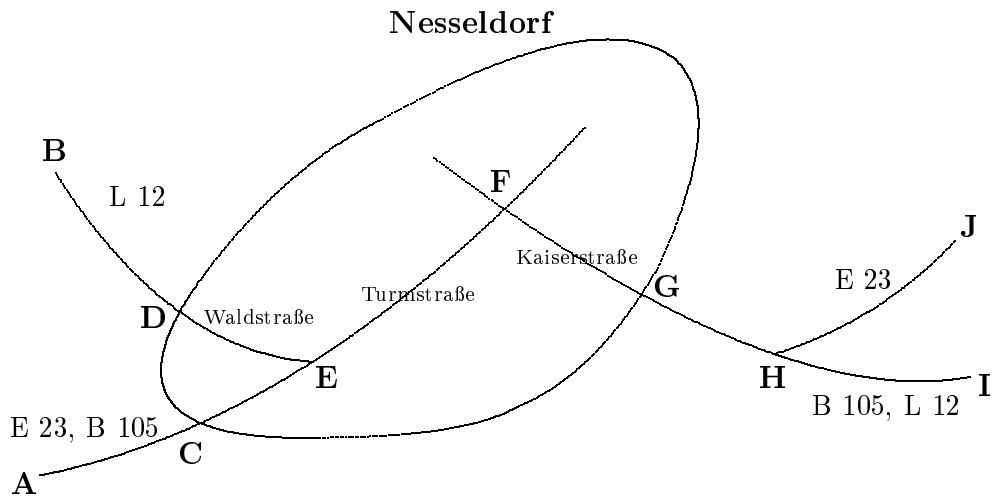
Abbildung 22: Die Struktur einer Komplexobjektart, bestehend aus zwei Objektarten

### 7.4.1 Komplexobjektart klassifizierte Straße

In Abschnitt 7.2 wurde bereits erwähnt, daß sich Redundanzen bei den DLM- und objektübergreifenden Attributen durch die Schaffung neuer Komplexobjektarten vermeiden lassen. Zur Verdeutlichung soll das in Abbildung 23 skizzierte Beispiel aus dem Straßenverkehr dienen<sup>8</sup>.

Die Bundesstraßen B 12 und B 105 sowie die Europastraße E 23 verlaufen zwischen den Punkten C und G auf dem selben Straßenkörper. Innerhalb dieser Strecke durchquert die Straße eine Ortschaft. Die Straßenabschnitte innerhalb der Ortschaft besitzen Straßennamen, alle anderen führen lediglich eine Kurzbezeichnung. Ohne die Verwendung von

<sup>8</sup> Die Abbildung dient nur der Verdeutlichung des Sachverhaltes und stellt kein real existierendes Szenarium dar.



OJ-Nr.	Objektteile	Widmung	intern. Bed.	Geogr. Name	Kurzbezeichnung
1	A - C	Bundesstraße	Europastraße	keiner	B 105, E 23
2	B - D	Landesstraße	keine	keiner	L 12
3	C - E	Bundesstraße	Europastraße	Turmstraße	B105, E 23
4	D - E	Landesstraße	keine	Waldstraße	L 12
5	E - F	Bundesstraße	Europastraße	Turmstraße	L 12, B 105, E 23
6	F - G	Bundesstraße	Europastraße	Kaiserstraße	L 12, B 105, E 23
7	G - H	Bundesstraße	Europastraße	keiner	L 12, B 105, E 23
8	H - I	Bundesstraße	keine	keiner	L 12, B 105
9	H - J	keine	Europastraße	keiner	E 23

Abbildung 23: Beispiel für ein Straßennetz ohne Komplexobjekte

Objektart	OJ-Nr.	Objektteile	Widmung	intern. Bed.	Geogr. Name	Kurzbez.
Straße	1	A - C	keine	keine	keiner	keiner
	2	B - D	keine	keine	keiner	keiner
	3	C - E	keine	keine	Turmstraße	keiner
	4	D - E	keine	keine	Waldstraße	keiner
	5	E - F	keine	keine	Turmstraße	keiner
	6	F - G	keine	keine	Kaiserstraße	keiner
	7	G - H	keine	keine	keiner	keiner
	8	H - I	keine	keine	keiner	keiner
	9	H - J	keine	keine	keiner	keiner
klassif.	1	2,4,5,6,7,8	Landesstraße	keine	keiner	L 12
Straße	2	1,3,4,5,6,7,8	Bundesstraße	keine	keiner	B 105
(komplex)	3	1,3,4,5,6,7,9	keine	Europastraße	keiner	E 23

Tabelle 9: Beispiele für die Komplexobjektart klassifizierte Straße (komplex)



Komplexobjekten und unter Beachtung aller Objekt- und Objektteilbildungsregeln (siehe dazu Abschnitt 8), ergibt sich die in Abbildung 23 dargestellte Objektstruktur.

Wie leicht zu erkennen ist, werden bei diesem Verfahren viele Informationen wie geographischer Name und Kurzname redundant gespeichert, andere, wie die Eigenschaft Landesstraße des Objektes 5, sind nicht mehr explizit abgelegt. Die ATKIS-Dokumentation [ATKIS] stellt diesbezüglich folgende Richtlinie auf:

‘Straße‘ und ‘Weg‘ gehören zu den Objektarten, die im allgemeinen in mehrere Abschnitte auf Objektebene aufgeteilt werden müssen. Um das Gesamtobjekt bilden zu können, müssen die einzelnen Objekte über eindeutige Namen (z.B. A3) selektierbar sein, das heißt es wird kein komplexes Objekt gebildet.

Dieser Grundsatz steht im Widerspruch zur Forderung nach redundanzfreier Speicherung aller attributiven Informationen und erschwert damit eine einfache und korrekte Fortführung. Eine mögliche Lösung ist die Schaffung einer neuen Komplexobjektart *klassifizierte Straße (komplex)*<sup>9</sup>, welche jeweils alle Objekte einer Europastraße, Autobahn, Bundesstraße oder analoges beinhaltet. Diese Komplexobjektart sollte neben dem geographischen, dem Zweit- und dem Kurznamen die Attribute IBD (internationale Bedeutung) und WDM (Widmung) enthalten und Objekte der Objektarten 3101 (Straße), 3102 (Weg), 3103 (Platz) und 3104 (Straße (komplex)) vereinen. Werden die genannten Attribute auf Komplexobjektebene gefüllt, so bleiben ihre Entsprechungen auf Objektebene unbelegt. Diese Strategie ist zur Vermeidung von Redundanzen notwendig, erschwert aber gleichzeitig die Recherche. Dies stellt neben der etwas schwierigeren Integritätskontrolle den größten Nachteil des vorgeschlagenen Verfahrens dar, wird jedoch auch bei regulären ATKIS-Objektarten praktiziert (siehe z.B. Verwaltungseinheit (komplex) (7103)).

Tabelle 9 zeigt anhand des Beispiels aus Abbildung 23 die Bildung von komplexen Straßenobjekten.

#### 7.4.2 Weitere IfAG-interne Komplexobjektarten

Die ATKIS-konforme Objektbildung bei den Objektgruppen Schienenverkehr (3200), Wasserflächen (5100), Verwaltungsgebiete (7100), Geographische Gebietseinheiten (7200), Schutzgebiete (7300) und Gefahrengebiete, sonstige Sperrgebiete (7400) sowie der Objektart Ortslage (2101) bedingt ebenfalls Redundanzen ähnlich den im vorangegangenen Abschnitt beschriebenen. Zusätzlich treten bei flächenhaften Objektarten (Gebiete und Ortslagen) Redundanzen auf geometrischer Ebene auf, da diesen nach ATKIS-Modellierungsregeln eine eigene Geometrie unabhängig von den Flächen der sachlich zugrundeliegenden Objektarten

---

<sup>9</sup> Nicht zu verwechseln mit der ATKIS-Komplexobjektart 3104 Straße (komplex)

zugeordnet wird. Eine Problemlösung durch Einführung neuer Komplexobjektarten ist auch hier möglich.

Eine weitere Quelle vermeidbarer Redundanzen sind die administrativen Einheiten (Verwaltungsgebiete). Die Verwaltungsgebiete der Bundesrepublik lassen sich laut [End95] trotz unterschiedlicher Namensgebung in den einzelnen Bundesländern jeweils einer der in Tabelle 10 aufgeführten Ebenen zuordnen. Die aktuelle Fassung der ATKIS-Dokumentation sieht die Erfassung jeder dieser Ebenen mit jeweils eigener Geometrie vor. Die hierarchischen Beziehungen zwischen den Ebenen sollen nicht verwaltet werden. Eine solche Vorgehensweise ist im Sinne einer effizienten und sicheren Fortführung nicht zu empfehlen. Statt dessen bietet sich die Schaffung von fünf Komplexobjektarten für die fünf höchsten Ebenen an, wobei Objekte eine Komplexobjektart immer aus den Objekten der niedrigeren Hierarchieebenen bestehen. Die Geometrie der administrativen Einheiten wird ausschließlich auf Gemeindeebene geführt, wofür die Objektart Verwaltungseinheit (7101) beibehalten werden kann.

Ebene	Rheinland-Pfalz	Niedersachsen	Brandenburg
1 Staat	—	—	—
2 Land	Land	Land	Land
3 Regierungsbezirk	Regierungsbezirk	Regierungsbezirk	—
4 Kreis	Kreis	Landkreis	Kreis
5 Verwaltungsgemeinschaft	Verbandsgemeinde	Samtgemeinde	Amt
6 Gemeinde	Ortsgemeinde	Gemeinde	Gemeinde

Tabelle 10: Die administrative Hierarchie in der BRD mit Beispielen [End95]

### 7.4.3 Bemerkungen zu den IfAG-internen Komplexobjektarten

Im Gegensatz zu den in Abschnitt 6.1.1 beschriebenen Objektarten wurden die IfAG-internen Komplexobjektarten noch nicht in die praktische Erprobung übernommen. Die Entscheidung, ob und wie diese Änderungen an der DLM-Datenstruktur vorgenommen werden sollten, ist von weiterführenden Arbeiten zu klären. Die an dieser Stelle eingeführten Lösungsansätze sind lediglich Empfehlungen für die weitere Verbesserung des DLM-Datenmodells.

## 7.5 DLM-Datenmodell im Überblick

In den letzten Abschnitten wurden die verschiedenen Komponenten der DLM-Datenstruktur vorgestellt und erläutert. Diese fügen sich zu der Struktur in Abbildung 24 zusammen. Die Abbildung zeigt die Struktur für eine Objektart eines DLM inklusive der Verbindungen zu

anderen Objektarten desselben und anderer DLM. In der Abbildung ist für die Beziehung *liegt\_über* eine von der Norm abweichende Darstellung verwendet worden. Dies resultiert aus der in Abschnitt 7.3.1 beschriebenen, vom relationalen Datenmodell abweichenden Modellierung dieser Beziehung. Die gestrichelten Linien stehen für optionale Beziehungen zwischen Entitätsmengen.

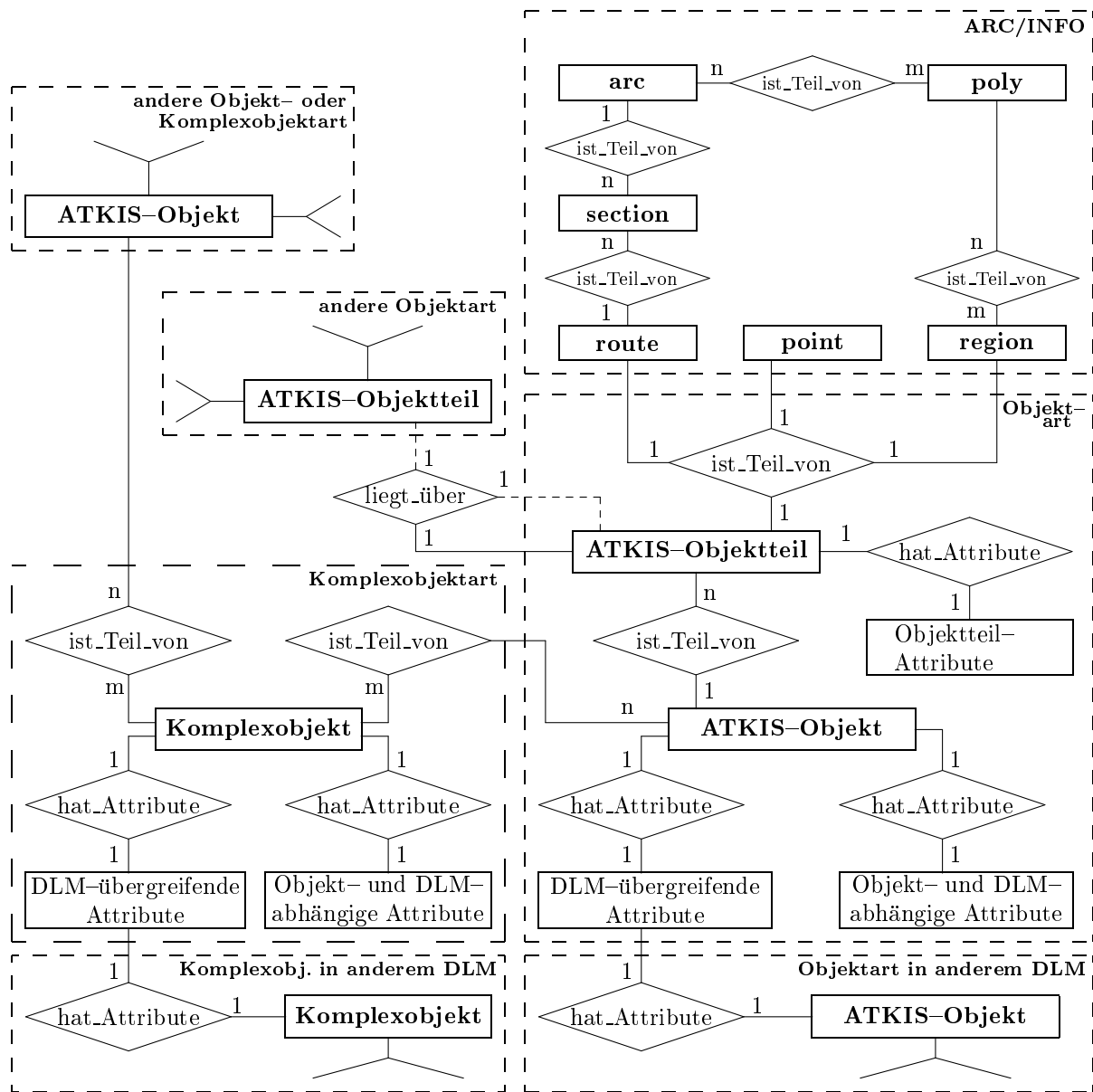


Abbildung 24: Entity/Relationship-Diagramm des DLM-Datenmodells mit komplexem Geometriemodell für eine Objektart

## 7.6 Implementation

Im Rahmen dieser Diplomarbeit wurde eine Prototypimplementation des vorgeschlagenen ATKIS-Datenmodells geschaffen. Dafür wurden die Module ARC, ARCDIT, DATABASE

INTEGRATOR sowie die Makrosprache AML (Arc Macro Language) des GIS ARC/INFO und das RDBMS Informix genutzt. Während dieser Arbeit entstanden die in Anhang B beschriebenen Tools.

Für diese Implementation war es notwendig, eine Reihe von Konventionen für verwendete Namen, Schlüssel und Datentypen aufzustellen. Außerdem mußten Festlegungen getroffen werden, wie Informationen aus den DST zu extrahieren und für verschiedene Operationen zu nutzen sind.

Der Aufbau der Tabellen der ATKIS-Datenbank ist in Tabelle 11 beschrieben und in Abbildung 25 dargestellt. Dabei sind neben Tabellen wie *liegt über*, die nur einmal existieren, die invarianten Teile der Tabellen beschrieben, die für jede Objekt- oder Komplexobjektart einmal vorhanden sind. Die varianten Teile sind gekennzeichnet und können aus den DST gewonnen werden.

Die Namen der Tabellen für Objektarten und Komplexobjektarten setzen sich zusammen aus einem konstanten Teil wie *objekt\_* oder *att1\_*, einem Kürzel für das DLM (*a*, *b* oder *c*) und dem vierstelligen Objektart- bzw. Objektartkode. Eine Ausnahme bilden die vom DLM unabhängigen Tabellen *att1\_XXXX*. Hier wird auf das DLM-Kürzel im Namen verzichtet.

Die variablen Teile der Tabellenstrukturen können vollständig aus den DST extrahiert werden. Dabei kann wie folgt vorgegangen werden. Zuerst wird der Name der Tabelle aus dem Grundnamen, dem gewünschten DLM und der Objektart zusammengesetzt. Dann werden unter Verwendung der Spalten *oar*, *dln* und *klasse* der Tabelle *hat\_attribut* alle für diese Tabelle relevanten Attribute aus den DST gewonnen. Dabei ist zu beachten, das z. B. für eine Tabelle des DLM 200 alle Attribute mit den *dln*-Vermerken *b*, *c*, *d* und *fgültig* sind. Der Inhalt der Spalte *attribut* in *hat\_attribut* ist gleichzeitig der Name der Spalte in der betrachteten Tabelle der ATKIS-Datenbank. Alle interessanten Informationen wie Datentyp, zugelassene Wertcodes u. v. m. lassen sich nun leicht unter Verwendung des Schlüsselattributes *attribut* aus den DST gewinnen.

## 7.7 weitere Etappen

Die hier vorgestellte Implementation verwirklicht nur einen Teil des in dieser Arbeit entwickelten DLM-Datenmodells. Um die effiziente Erfassung und Fortführung von ATKIS-Daten zu ermöglichen, sind zunächst die im nachfolgenden genannten Schritte notwendig:

- Vervollständigung und Fertigstellung der Tools für die Überführung von vorhandenen DLM-Objektteilen in die neue Datenstruktur
- Fertigstellung des Tools *atkisobj* für die Schaffung und Manipulation von DLM-Objekten

Gruppe	Tabelle	Spalten	Bemerkung
allgemein	liegt_ueber	tabarc1 char(15) idarc1 char(10) tabarc2 char(15) idarc2 char(10)	OAR des ersten Objektteiles unechter Fremdschlüssel (erster OT) OAR des zweiten Objektteiles unechter Fremdschlüssel (zweiter OT)
Objektarten	objekt_yxxxx <sup>a</sup>	objektkey char(10) datum_akt date herkunft char(2) XXX <sup>b</sup> char( <i>n</i> ) <sup>c</sup>	Primärschlüssel Aktualitätsdatum Quellennachweis Objektabhängige Attribute
	teil_yxxxx	otkey char(10) objektkey char(10) datum_neu date datum_aen date datum_loe date XXX char( <i>n</i> )	Primärschlüssel Fremdschlüssel zu objekt_yxxxx Eintragsdatum Änderungsdatum Löschdatum Objektteilattribute
	att1_xxxx	key200 char(10) key1000 char(10) XXX char( <i>n</i> )	Fremdschlüssel zu objekt_bxxxx Fremdschlüssel zu objekt_cxxxx DLM-übergreifende Attribute
Komplex- objektarten	objekt_y_xxxx	kobjektkey char(10) datum_akt date herkunft char(2) XXX char( <i>n</i> )	Primärschlüssel Aktualitätsdatum Quellennachweis Objektabhängige Attribute
	teil_yxxxx_zzzz <sup>d</sup>	kotkey char(10) kobjektkey char(10) datum_neu date datum_aen date datum_loe date	Primärschlüssel Fremdschlüssel zu kobjekt_yxxxx Eintragsdatum Änderungsdatum Löschdatum
	att1_xxxx	key200 char(10) key1000 char(10) XXX char( <i>n</i> )	Fremdschlüssel zu kobjekt_bxxxx Fremdschlüssel zu kobjekt_cxxxx DLM-übergreifende Attribute

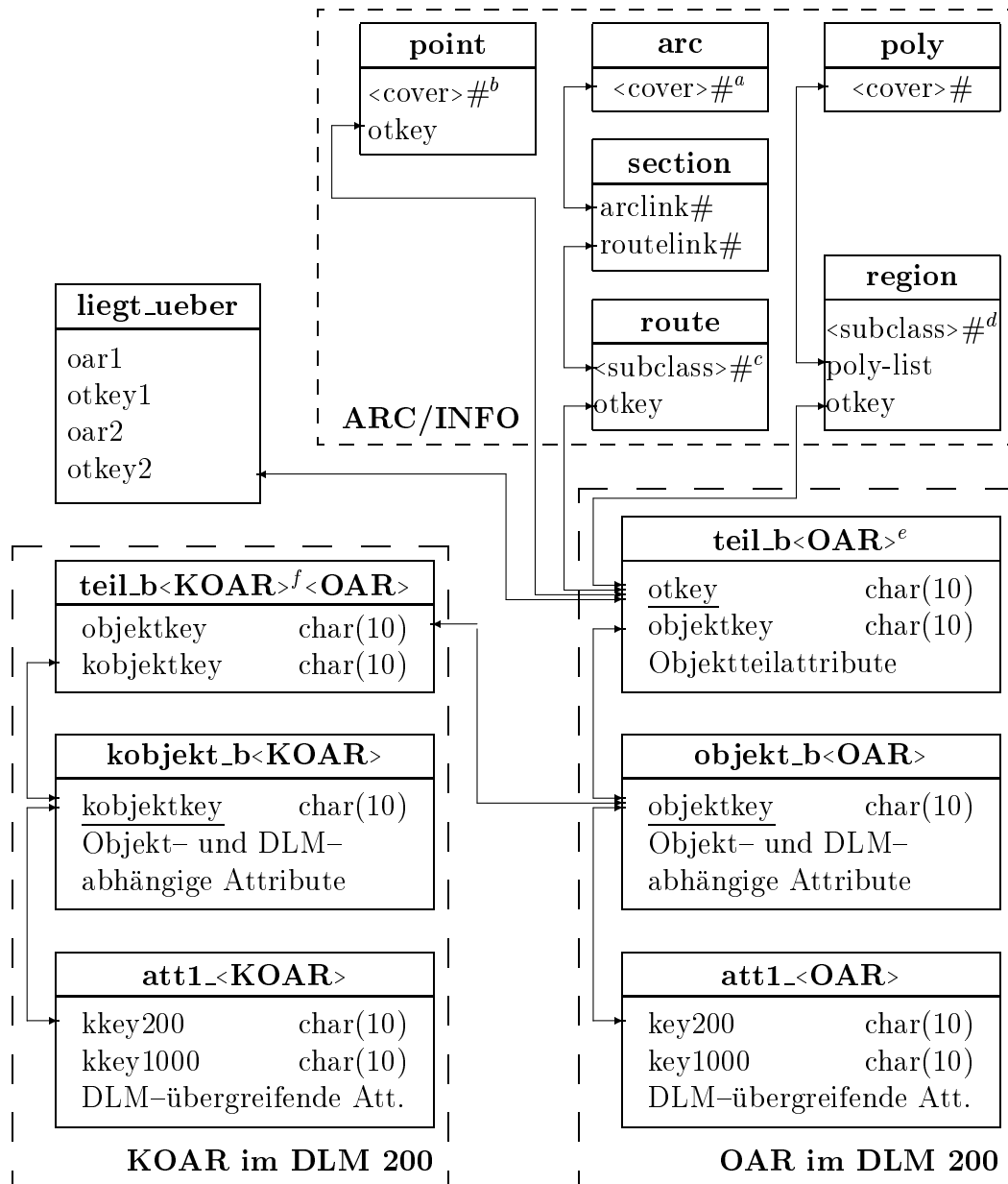
<sup>a</sup> Das *y* steht für einen der Buchstaben a, b oder c, entsprechend der Zugehörigkeit der Tabelle zum DLM 25, 200 oder 1000, während *xxxx* durch die Objektart zu ersetzen ist.

<sup>b</sup> XXX steht für den dreibuchstabigen Attributcode.

<sup>c</sup> Der Datentyp richtet sich nach der Art des Attributes. Es sind Zeichenketten verschiedener Länge (*n*) und Integer möglich.

<sup>d</sup> Mit *xxxx* wird die Komplexobjektart ausgewiesen, *zzzz* ist die Objektart, deren Objekte das Komplexobjekt bilden.

Tabelle 11: Die Tabellen der ATKIS-Datenbank



<sup>a</sup> Für <cover> ist der Name des Geometrie-covers einzusetzen.  
<sup>b</sup> Für <cover> ist der Name des Punkt-covers einzusetzen.  
<sup>c</sup> Für <subclass> ist der Name der verwendeten Routenklasse einzusetzen.  
<sup>d</sup> Für <subclass> ist der Name der verwendeten Regionenklasse einzusetzen.  
<sup>e</sup> Für <OAR> ist der vierstellige Objektartcode einzusetzen.  
<sup>f</sup> Für <OAR> ist der vierstellige Komplexobjektartcode einzusetzen.

Abbildung 25: Die Tabellen der ATKIS-Datenbank und ihre Beziehungen (aktueller Stand der Implementation)

- Schaffung eines Tools zur Erfassung und Manipulation von DLM-Objektteilen unter Nutzung des komplexen Geometriemodells (Regionen und Routen)
- Implementation der Strukturen für Komplexobjekte und Erweiterung der Funktionalität bestehender Tools für die Verwaltung von Komplexobjekten
- Spezifikation und Implementation der temporalen Komponente der DLM-Datenbank und der DST

## 8 Modellierungsregeln für ATKIS–Objekte

Dieser Abschnitt beinhaltet eine Aufstellung allgemeiner Objekt– und Objektteilbildungsregeln und die damit verbundenen Modellierungsvorschriften und Erfassungskriterien. Diese wurden teilweise aus der ATKIS–Dokumentation, dem Objektartenkatalog und den Modellierungsbeispielen gewonnen. Andere Regeln wurden aus kartographisch–topologischen Gründen aufgestellt, insbesondere in Fällen, in denen ATKIS nur unzureichende oder keine Aussagen trifft. Diese Regeln sind im Text besonders ausgewiesen. Des weiteren ergeben sich aus der Analyse der ATKIS–Datenstruktur einige Einschränkungen bzw. Vorschriften für die Modellierung von ATKIS–Objekten.

Regeln können allgemein, für ganze Objektgruppen oder für einzelne Objektarten gelten. Außerdem ist eine Unterscheidung zwischen den verschiedenen Modellierungsarten punkt–, linien– und flächenförmig notwendig.

Für jedes DLM und für jeden Objekttyp existieren Erfassungskriterien (siehe [ATKIS]), d.h. Vorschriften, ab welcher Ausdehnung ein reales Landschaftsobjekt als ATKIS–Objekt erfaßt wird. Unterschreiten Landschaftsobjekte die angegebene Mindestgröße, so sind sie einem geeigneten benachbarten Landschaftsobjekt zuzuschlagen. Dies kann dazu führen, daß mehrere zu kleine Objekte zu einem größeren zusammengefügt werden, welches dann die Kriterien erfüllt.

### 8.1 Attribute

Ein Attribut kann nur mit einem Attributwert belegt werden. Sind mehrere Attributwerte zutreffend, wird der am ehesten zutreffende, dominantere ausgewählt. Jedes Attribut ist mit einem Wert zu versehen, auch wenn der tatsächliche Wert noch nicht bekannt ist. In diesem Fall ist das Attribut mit 9998 ‘nach Quellenlage derzeit keine Zuweisung möglich’ bzw. bei Namen mit ‘NNNN’ zu füllen.

Ist der Wert 9999 ‘Sonstiges’ für ein Attribut aufgeführt, so kann er verwendet werden, wenn der tatsächliche Sachverhalt bekannt, jedoch keiner der anderen Werte zutreffend ist.

Es gibt Attribute, welche Sonderfälle beschreiben, und deshalb nur selten mit einem sinnvollen Wert belegt werden müssen. In diesen Fällen wurde auf eine explizite Formulierung der entsprechenden negativen Aussage verzichtet und der allgemeine Attributwert 9997 ‘Attribut trifft nicht zu’ geschaffen.

Die genannten Regeln gelten nicht für Attribute, die mit ihrem tatsächlichen numerischen Attributwert belegt werden. Kann das Attribut aufgrund des Objekttyps nicht sinnvoll gefüllt werden, so ist auch hier der Wert 9997 zu verwenden.



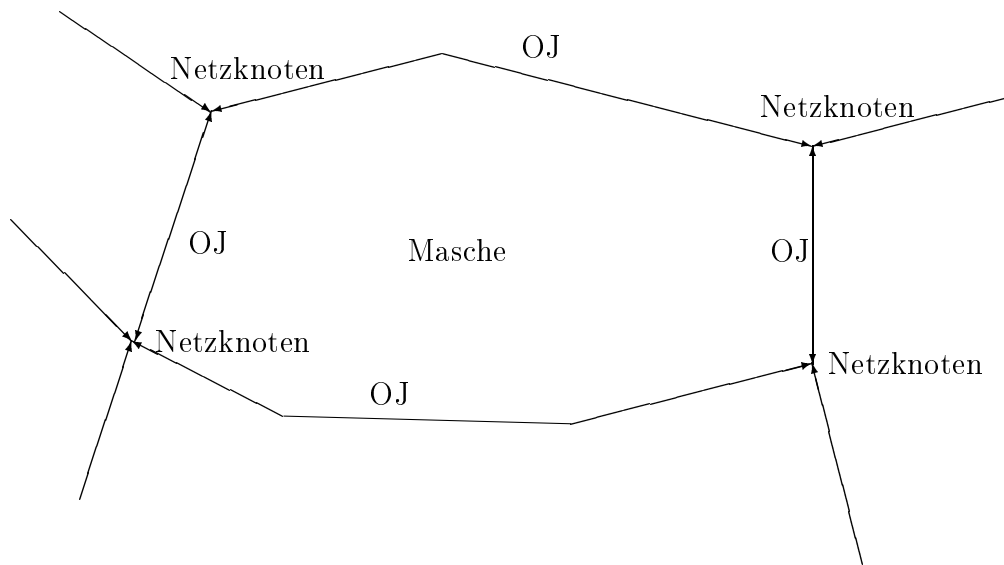


Abbildung 26: Beispiel für die Objektbildung in Netzen

## 8.2 Netzbildung

Die linienhaften Objekte der in Tabelle 12 aufgeführten Objektarten bilden das topologische Straßen-, das Eisenbahn- und das Gewässernetz. Dabei sind laut [ATKIS] an allen niveaugleichen Berührungspunkten innerhalb der Netze sogenannte Netzknnoten zu bilden, aber nur, wenn auch eine physische Vernetzung besteht, nicht jedoch bei Brücken, Überführungen oder ähnlichem. Berührungspunkte zwischen den Netzen sind nur in Ausnahmefällen, wenn zwischen den Netzen eine echte Verbindung besteht, als Knoten zu modellieren. Dies ist zum Beispiel bei Auto- und Eisenbahnfähren der Fall.

Bei Objektarten der Netze ist nach [ATKIS] an jedem Netzknnoten ein neues, eigenständiges Objekt zu bilden. Dies führt, wie in Abb. 26 zu sehen, zu einer Zerstückelung der Netze in Objekte, welche sich höchstens von Netzknnoten zu Netzknnoten erstrecken. Der sich dadurch ergebende Verlust von globaleren Zusammenhängen, z.B. im Straßennetz, kann durch die Bildung von Komplexobjekten, wie in Abschnitt 7.4.1 beschrieben, behoben werden.

Im Bedarfsfall kann das Stadtbahnnetz getrennt vom Eisenbahnnetz betrachtet werden. Dies ist laut [ATKIS] dann der Fall, wenn U-Bahn und Straßenbahn ein gemeinsames Schienennetz benutzen.

Durch die topologischen Netze wird die Landschaft in Maschen eingeteilt, welche durch die in Abschnitt 8.3.3 beschriebenen Maschenflächen gefüllt werden. Dabei sind auch Abschnitte der Landschaft als Maschen zu verstehen, bei denen das umgebende Polygon nicht ausschließlich durch Netzknnoten, sondern auch durch einfache geometrische Überschneidungen der Netze gebildet wird.

Art des Netzes	Objektarten dieses Netzes
Straßennetz	3101 Straße
	3102 Weg
	3403 Schifffahrtslinie mit FKT = 'Autofährverkehr'
Eisenbahnnetz	3201 Schienenbahn mit BKT = 'Eisenbahn'
	3403 Schifffahrtslinie mit FKT = 'Eisenbahnfährverkehr'
Stadtbahnnetz	3201 Schienenbahn mit BKT = 'Stadtbahn'
Gewässernetz	5101 Strom, Fluß, Bach
	5102 Kanal(Schifffahrt)
	5103 Graben, Kanal(Wasserwirtschaft)
	5104 Priel
	5105 Quelle
	5111 Meer
	5112 Binnensee, Stausee, Teich
5121 Watt	

Tabelle 12: Die topologischen Netze für die DLM 200 und 1000

### 8.3 Objektteilbildungsregeln

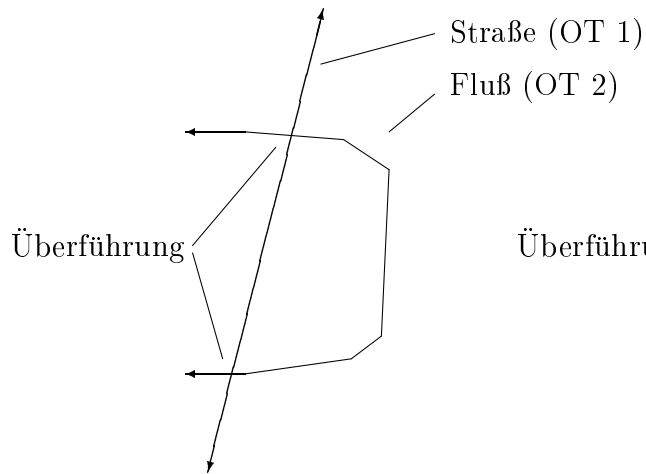
Ein Objektteil (OT) ist ein konkreter, geometrisch begrenzter, durch einheitliche Attribute und Relationen bestimmter Teil eines Objektes (OJ) [ATKIS]. Demzufolge wird immer dann, wenn sich ein Attributwert ändert, ein Attribut hinzukommt oder ein Attribut wegfällt, ein neuer OT gebildet. An den Knoten der in 8.2 beschriebenen topologischer Netze ist immer ein neuer OT zu bilden.

Wenn ein OT räumlich über oder unter einem anderen OT liegt, so ist diese Beziehung in Form einer Überführungsreferenz anzugeben. Diese binäre Relation verbindet jeweils einen ‚OT oben‘ und einen ‚OT unten‘. Hierbei werden nur in vertikaler Richtung direkt benachbarte OT referenziert. Liegt zum Beispiel eine Straße über einer Brücke und diese wiederum über einem Fluß, so werden nur genau diese beiden, aber nicht die Beziehung ‚Straße liegt über Fluß‘ abgelegt.

Bezüglich eines Objektteiles sind mehrere Referenzen möglich, solange die referenzierten OT verschieden sind. Tritt jedoch ein wie in Abbildung 27 skizzierter Fall auf, in welchem zwischen zwei OT an mehreren Stellen Referenzen gebildet werden müssen, so ist zumindest einer der beteiligten OT in geeigneter Weise in mehrere OT aufzuteilen. Auf diese Weise ist die Eindeutigkeit der Zuordnung der Referenz zu ihrem geometrischen Referenzpunkt sicherzustellen.

Im Verlaufe der topologischen Netze auftretende Brücken, Überführungen und Unter-

### Falsche Modellierung

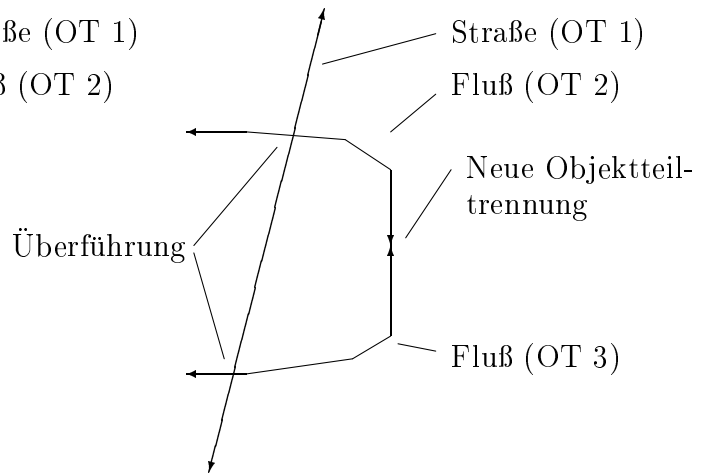


Referenztable

OT oben	OT unten
1	2
1	2

Referenzen scheinbar doppelt, keine eindeutige Zuordnung zu Geometrie möglich

### Korrekte Modellierung



Referenztable

OT oben	OT unten
1	2
1	3

Referenzen sind jetzt eindeutig zuzuordnen

Abbildung 27: Beispiel für die Aufteilung von Objektteilen bei mehrfachen Referenzen

führungen sind stets als eigenständige Objekte zu modellieren und nach den genannten Regeln mit den OT der Netze zu referenzieren. Dabei ist zu beachten, daß eine Brücke, Überführungen oder Unterführungen keine Begründung für die Bildung eines neuen OT bei den darüber oder darunter liegenden Objekten ist. Als Veranschaulichung hierfür soll Abbildung 28 dienen.

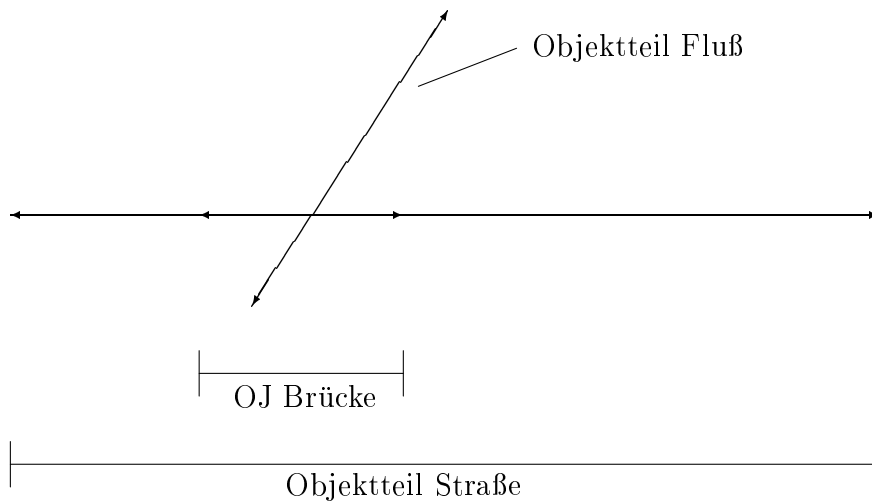


Abbildung 28: Beispiel für die Modellierung einer Brücke im Verlaufe einer Straße

### 8.3.1 punktförmige Objektteile

Eine Fläche wird durch ein Objektteil des Objekttyps punktförmig modelliert, wenn sich kein Punkt des Randes der Fläche signifikant, gemessen an der Modellgenauigkeit, von einem anderen unterscheidet. Die Lage des Objektteils ergibt sich aus dem Mittelpunkt der realen Objektteilfläche.

### 8.3.2 linienhafte Objektteile

Eine Fläche wird durch ein Objektteil des Objekttyps linienförmig modelliert, wenn der Rand im Rahmen der Modellgenauigkeit im wesentlichen durch zwei Parallelen ersetzt werden kann, und die Länge dieser Parallelen sehr viel größer als deren Abstand ist. Diese OT werden geometrisch durch ihre Mittelachse repräsentiert.

### 8.3.3 flächenhafte Objektteile

Eine Fläche wird durch ein Objektteil des Objekttyps flächenförmig modelliert, wenn sich Punkte des Randes im Rahmen der Modellgenauigkeit signifikant voneinander unterscheiden. Die Ränder einander benachbarter Flächen können immer dort als identisch angesehen

werden, wo sie sich im Rahmen der Modellgenauigkeit nicht signifikant voneinander unterscheiden.

Die Landschaft wird durch die topologischen Netze (siehe Tab. 12) in Maschen aufgeteilt. Diese werden dann zunächst mit Objektteilen der in Tabelle 15 aufgeführten Objektarten gefüllt. Diese Objektarten werden auch als Maschenobjekte bezeichnet. Dabei darf jeder Objektteil nur Maschenobjekte der in der Tabelle aufgeführten Objektarten überlagern. Eine vollständige Füllung aller Maschen mit Objektteilen ist nur beim DLM 25 gefordert.

Nicht zu verwechseln sind Objektarten der Maschenflächen mit den in Tabelle 13 aufgelisteten Objektarten der sogenannten Grundflächen. Diese sind Objektarten, die sich, um eine redundanzfreie Beschreibung der Erdoberfläche zu erreichen, nur in Ausnahmefällen überlagern dürfen. Alle anderen flächenhaften OT dürfen diese Grundflächen überlagern. Sie werden deshalb auch Überlagerungsflächen genannt (Tab. 14). Grundsätzlich ist also jede mit ATKIS-Objekten dieser beiden Gruppen belegte Fläche von einer Grundfläche und optional einer oder mehreren Überlagerungsflächen okkupiert. Eine Überlagerungsfläche kann demzufolge nie separat ohne zugehörige Grundfläche auftreten.

Eine Sonderstellung nehmen die Objektarten Tagebau, Grube, Steinbruch (3201), Halde, Aufschüttung (2302), Hafenbecken (3402) und Binnensee, Stausee, Teich (5112) ein. Diese können sowohl Grund- als auch Überlagerungsflächen sein. Des weiteren dürfen Ortslage (2101), Hafen (3401) sowie alle Objektarten des Objektbereiches Gebiete (7xxx) alle Objektarten überlagern. Diese sind aufgrund ihrer Sonderstellung weder den Grund-, noch den Überlagerungsflächen zuzuordnen.

## 8.4 Objektbildungsregeln

Ein neues Objekt wird gebildet, wenn Objekte verschiedener Objektarten aneinandergrenzen, oder sich ein Objektattribut oder der Objekttyp ändern. Objektattribute sind die im OK verzeichneten *hervorgehobenen Attribute im Sinne der Objektbildung* und alle Namen.

Die im OK geforderte Objektbildung an Landesgrenzen wird für die DLM 200 und 1000 außer Kraft gesetzt, da sie für diese Maßstabsbereiche wenig sinnvoll erscheint.

Aus den in Abschnitt 8.3.3 und in Tabelle 15 aufgelisteten Maschenobjektteilen wird jeweils ein Objekt gebildet. D.h. jedes Maschenobjekt besteht prinzipiell aus einem Objektteil.

## 8.5 Regeln zu Objektarten

In diesem Abschnitt werden exemplarisch anhand des Objektbereiches Siedlung die während dieser Arbeit in enger Zusammenarbeit mit Mitarbeitern des IfAG entstandenen Modellierungsregeln vorgestellt. Es handelt sich dabei um detailliertere Anforderungen an einige

OAR	Objektart	OAR	Objektart
2111	Wohnbaufläche	4102	Grünland
2112	Industrie- und Gewerbefläche	4103	Gartenland
2113	Fläche gemischter Nutzung	4104	Heide
2114	Fläche besonderer funktionaler Prägung	4105	Moor, Moos
2201	Sportanlage	4106	Sumpf, Ried
2202	Freizeitanlage	4107	Wald, Forst
2213	Friedhof	4108	Gehölz
2301	Tagebau, Grube, Steinbruch	4109	Sonderkultur
2302	Halde, Aufschüttung	4110	Brachland
3101	Straße	4120	Vegetationslose Fläche
3105	Straßenkörper	4199	Fläche z.Z. unbestimmbar
3201	Schienenbahn	5101	Strom, Fluß, Bach
3204	Bahnkörper	5102	Kanal (Schifffahrt)
3402	Hafenbecken	5103	Graben, Kanal (Wasserwirtschaft)
3501	Bahnhofsanlage	5111	Meer
4101	Ackerland	5112	Binnensee, Stausee, Teich
		5302	Talsperre, Wehr

Tabelle 13: Objektarten der Grundflächen

OAR	Objektart	OAR	Objektart
2121	Bergbaubetrieb	2228	Campingplatz
2122	Abfalldeponie	2229	Autokino, Freilichtkino
2123	Raffinerie	2301	Tagebau, Grube, Steinbruch
2125	Lager, Depot	2302	Halde, Aufschüttung
2126	Kraftwerk	3301	Flughafen
2127	Umspannstation	3302	Flugplatz, Landeplatz
2128	Förderanlage	3402	Hafenbecken
2133	Heizwerk	4111	Nasser Boden
2135	Abfallbeseitigungsanlage	5112	Binnensee, Stausee, Teich
2221	Stadion		

Tabelle 14: Objektarten der Überlagerungsflächen

OAR	Objektart	Überlagerung erlaubt
2111	Wohnbaufläche	
2112	Industrie- und Gewerbefläche	
2113	Fläche gemischter Nutzung	
2114	Fläche besonderer funktionaler Prägung	
2121	Bergbaubetrieb	2111,2112,2113,2114
2122	Abfalldeponie	2111,2112,2113,2114
2123	Raffinerie	2111,2112,2113,2114
2125	Lager, Depot	2111,2112,2113,2114
2126	Kraftwerk	2111,2112,2113,2114
2127	Umspannstation	2111,2112,2113,2114
2128	Förderanlage	2111,2112,2113,2114
2133	Heizwerk	2111,2112,2113,2114
2135	Abfallbeseitigungsanlage	2111,2112,2113,2114
2201	Sportanlage	
2202	Freizeitanlage	
2213	Friedhof	
2221	Stadion	2201,2202
2228	Campingplatz	2201,2202
2229	Autokino, Freilichtkino	2201,2202
3301	Flughafen	
3302	Flugplatz, Landeplatz	
4104	Heide	
4105	Moor, Moos	
4106	Sumpf, Ried	
4107	Wald, Forst	
4109	Sonderkultur	
4111	Nasser Boden	
4120	Vegetationslose Fläche	
4199	Fläche, z.Z. unbestimmbar	

Tabelle 15: Maschenobjektarten

Objektarten, Objektgruppen und Objektbereiche sowie Ausnahmen zu den bisher aufgestellten Regeln.

Die Objektarten 2111 bis 2114 der Objektgruppe 'Baulich geprägte Flächen' werden für das DLM 25 unabhängig von ihrer Größe erfaßt, also auch einzeln stehende Häuser oder Anwesen, selbst wenn diese weit unter der minimalen Erfassungsgröße liegen. Für die DLM 200 und 1000 gelten die in den OK angegebenen Mindestgrößen. 'Baulich geprägte Flächen' sind alle ständig bewohnten oder regelmäßig von Menschen genutzte Einrichtungen, nicht jedoch untergeordnete Gebäude wie Schuppen, Friedhofsgebäude, Jagdhäuser oder Wochenendhäuser. Befindet sich ein unter der Mindestgröße liegender OT neben anderen OT der Objektgruppe 'Baulich geprägte Flächen', so ist er einem dieser OT, und zwar dem mit den vergleichsweise ähnlichsten Eigenschaften, zuzuschlagen. Es ist jedoch verboten, diese OT einer anderen Objektgruppe zuzuordnen.

Jeder OT der Objektgruppe 'Baulich geprägte Flächen' ist prinzipiell einer der Objektarten 2111 bis 2114 zuzuordnen, die Objektarten 2121 bis 2135 müssen immer einer dieser Objektarten überlagert werden.

Die Objektart Ortslage erfordert noch einige eingehendere Bemerkungen. Hierunter verstehen wir eine im Zusammenhang bebaute Fläche von mindestens ca. 10 ha oder 10 Anwesen (im DLM 200). Dabei sind nicht nur alle baulich geprägten Flächen sondern auch alle in engem räumlichen oder funktionellen Zusammenhang stehenden Flächen beinhaltet, insbesondere wenn diese Flächen sonst Inseln innerhalb der Ortslage bilden würden. Der Umring einer Ortslage bildet stets einen geschlossenen Linienzug, Inseln innerhalb der Ortslage sind bis auf Vegetationsflächen nicht erlaubt.

Der Name einer Ortslage ist der der jeweils höchsten zutreffenden Hierarchiestufe (Gemeinde, Gemeindeteil, Wohnplatz).

Die Objektart Ortslage wird auch erfaßt, wenn zwar die Mindestgröße unterschritten wird, jedoch mindestens eines der folgenden Kriterien erfüllt ist.

Die Ortslage

- gibt einer Gemeinde den Namen
- beherbergt einen Verwaltungssitz
- enthält einen Bahnhof
- gibt einem Bahnhof den Namen
- gibt einem Verkehrsknoten den Namen
- verknüpft in ihrem Inneren Fernstraßen
- gibt einem Flughafen den Namen



- ist der Ausgangspunkt von Schifffahrtslinien
- überführt in ihrem Inneren Fernverkehr über Gewässer

Das Objekt ist in diesem Fall punktförmig zu modellieren.

### **8.5.1 Regeln zu weiteren Objektarten**

Ähnlich den im vorangegangenen Abschnitt vorgestellten Regeln für den Objektbereich Siedlung wurden am IfAG Regelwerke für alle weiteren Objektbereiche erarbeitet. Da diese für die Aufgabenstellung dieser Arbeit jedoch nur von untergeordnetem Interesse sind, soll hier auf eine Dokumentation verzichtet werden.

## 9 Zusammenfassung

In den vorangegangenen Kapiteln wurde eine Einführung in die Themenkomplexe ATKIS und temporale Datenmodelle/Metadatenbanken gegeben. Die während dieser Arbeit erfolgte Analyse von ATKIS und speziell den OK wurde dokumentiert und führte schließlich zur Schaffung des Metainformationssystems DST. Aufbauend auf diesen Grundlagen wurde ein DLM-Datenmodell entwickelt und an die technischen Möglichkeiten des IfAG angepaßt. Weiterhin wurden Vorschläge einer temporalen Erweiterung dieses Modells unterbreitet. Anhand eines Beispiels wurden die aus den OK und allgemeinen kartographischen Regeln und technologischen Einschränkungen gewonnenen Objektbildungsregeln für ATKIS-Objekte vorgestellt.

Im Laufe der Arbeit wurden folgende wesentliche Erkenntnisse gewonnen und Resultate erzielt:

- Das Amtliche Topographisch-Kartographische Informationssystem (ATKIS) ist ein Vorhaben der Landesvermessungsverwaltungen der Bundesrepublik Deutschland zum Aufbau Digitaler Landschaftsmodelle und Digitaler Kartographischer Modelle.
- Für das Digitale Landschaftsmodell (DLM) werden auf Grundlage der Modelltheorie Landschaftsobjekte nach Form, Lage und topologischen Relationen abgegrenzt und erfaßt, den im Objektartenkatalog aufgeführten Objektarten zugeordnet, durch Attribute beschrieben, verschlüsselt und gespeichert.
- Die Abgabe von ATKIS-Daten an Dritte erfolgt über die Einheitliche Datenbankschnittstelle (EDBS) durch das Verfahren Beziehersekundärnachweis (BZSN). Nach einmaliger Abgabe der Erstausrüstung wird nur noch die Inkrementelle Fortführungsinformation abgegeben.
- Die Gewinnung der Inkrementellen Fortführungsinformation erfordert ein Datenmodell, welches die Speicherung von Änderungsoperationen und älterer Datenbankzustände ermöglicht. Spezielle temporale Datenmodelle können die geforderten Bedingungen erfüllen.
- Zeitbezogene Entwicklungen bei geometrischen und attributiven Geodaten lassen sich vereinfacht mit zustandserhaltenden Geschichten beschreiben.
- Durch die Unterscheidung flüchtiger, Arbeits- und freigegebener Versionen werden Erfassungs- und Fortführungsprozesse von Geodaten gut unterstützt.

- Bitemporale Datenmodelle unterstützen Gültigkeits- und Aufzeichnungszeiten. Eine bitemporale Erweiterung des Relationenmodells erlaubt die Verwendung herkömmlicher RDBMS zur Abbildung der temporalen Komponente.
- Ein temporales DLM-Datenmodell macht eine Erweiterung der verwendeten Datenbankansprache unumgänglich. Die zusätzlichen Sprachelemente können vollständig mit Elementen von SQL nachgebildet werden. Die Generierung kann durch einen Preprozessor übernommen werden.
- Die komplexe Struktur des DLM-Datenmodells erfordert die Verwaltung durch ein Metainformationssystem. Die DLM-Datenstruktur befindet sich noch in der Entwicklung und ist in gewissen Grenzen veränderlich.
- Metainformationssysteme können mit gewissen Einschränkungen temporal erweitert werden. Beschreibt ein temporales Metainformationssystem temporale Daten, so sind starke Restriktionen für Veränderungen der Metadaten notwendig. Es können jedoch alle für die DLM-Datenstruktur zu erwartenden Veränderungen beschrieben werden.
- Das Metainformationssystem Datenstrukturtabellen (DST) enthält den für das DLM-Datenmodell wesentlichen Inhalt der drei Objektartenkataloge sowie zusätzliche technologische Informationen. Es bildet die Grundlage aller automatischen und interaktiven Werkzeuge für die DLM-Datenbank.
- Die Analyse der OK brachte eine Reihe von Unstimmigkeiten zutage. Ein Teil konnte in Zusammenarbeit mit dem zuständigen Mitarbeiter des IfAG bereits ausgeräumt werden.
- Die Verwaltung der geometrischen DLM-Daten unter ARC/INFO ist auf Grundlage des einfachen oder des komplexen Geometriemodells möglich. Das komplexe Modell ist im Hinblick auf die Fortführung der DLM-Daten günstiger, stößt jedoch momentan noch auf technische Schwierigkeiten.
- Attributive DLM-Daten können durch die Zuordnung zu drei getrennt gespeicherten Attributklassen nahezu redundanzfrei gehalten werden. Die physische Speicherung erfolgt gemeinsam mit den Objekt- und Objektteilstrukturen außerhalb von ARC/INFO unter einem herkömmlichen RDBMS.
- Komplexen hierarchische Zusammenhänge, zum Beispiel im Straßensystem oder bei Verwaltungseinheiten, lassen sich vorteilhaft durch die Bildung neuer Komplexobjektarten abbilden.

- Die durch die ATKIS–Konzeption vorgegebenen, durch die DLM–Datenstruktur bedingten und aus kartographisch–topologischen Gründen aufgestellten Regeln für die Objekt– und Objektteilbildung, die Bildung topologischer Netze und die Bildung topologischer Referenzen ergeben einen umfangreichen, Modellierungsregeln für ATKIS–Objekte genannten Katalog.

# Tabellenverzeichnis

1	Die Namen der Feature Attribute Tables . . . . .	20
2	Bezeichnungsfehler bei Objektarten . . . . .	49
3	Beispiele für die Verwendung von FLA und FLB . . . . .	51
4	Beispiele für die mehrdeutige Verwendung von Attributkürzeln . . . . .	52
5	Beispiele für Zusammenfassungsfehler bei Attributwerten . . . . .	52
6	Beispiele für inhaltliche Fehler bei Attributwerten . . . . .	53
7	Beispiele für fehlende Attribute und Attributwerte . . . . .	54
8	Attributklassen und die Anzahl ihrer Elemente . . . . .	60
9	Beispiel für klassifizierte Straße . . . . .	65
10	Die administrative Hierarchie in der BRD . . . . .	67
11	Die Tabellen der ATKIS-Datenbank . . . . .	70
12	Die topologischen Netze für die DLM 200 und 1000 . . . . .	75
13	Objektarten der Grundflächen . . . . .	79
14	Objektarten der Überlagerungsflächen . . . . .	79
15	Maschenobjektarten . . . . .	80
16	Objektattribute und ihre Zuordnung zu Attributklassen, Teil 1 . . . . .	xxxix
17	Objektattribute und ihre Zuordnung zu Attributklassen, Teil 2 . . . . .	xxxix
18	Objektattribute und ihre Zuordnung zu Attributklassen, Teil 3 . . . . .	xxxix
19	Objektattribute und ihre Zuordnung zu Attributklassen, Teil 4 . . . . .	xxxix
20	Objektattribute und ihre Zuordnung zu Attributklassen, Teil 5 . . . . .	xxxix
21	Objektattribute und ihre Zuordnung zu Attributklassen, Teil 6 . . . . .	xxxix
22	Objektattribute und ihre Zuordnung zu Attributklassen, Teil 7 . . . . .	xxxix
23	Objektattribute und ihre Zuordnung zu Attributklassen, Teil 8 . . . . .	xxxix
24	Komponenten der Datenstrukturtabellen, Teil 1 . . . . .	xli
25	Komponenten der Datenstrukturtabellen, Teil 2 . . . . .	xlii

# Abbildungsverzeichnis

1	Komponenten der Diplomarbeit . . . . .	1
2	Zusammenhang Primär-, Sekundär-, und Tertiärmodell der Landschaft . . .	5
3	Informationsfluß in ATKIS . . . . .	6
4	Die ATKIS-Komponenten und ihre Beziehungen . . . . .	8
5	Auszug aus dem OK 200 . . . . .	11
6	Struktur des DLM-Datenmodells . . . . .	12
7	Struktur des DKM-Datenmodells . . . . .	13
8	Abgabe und Fortführung der ATKIS-Daten . . . . .	14
9	Die wichtigsten Geometrietypen in ARC/INFO und ihre Beziehungen . . . .	19
10	Beispiel für eine zustandserhaltende Geschichte . . . . .	25
11	Beispiel für eine zustandsverändernde Geschichte . . . . .	26
12	Beispiel für eine anleitbare Geschichte . . . . .	27
13	Beispiel für eine ereignisorientierte Geschichte . . . . .	28
14	Beispiel für eine bitemporale Objektgeschichte . . . . .	31
15	Beispiel einer Versionenhierarchie . . . . .	32
16	Entity/Relationship-Modell der Datenstrukturtabellen . . . . .	45
17	Die Tabellen der Datenstrukturtabellen und ihre Beziehungen (Version 1.8) .	48
18	Ausgabeseite der AML-Routine <code>db_report.aml</code> . . . . .	50
19	Einfaches Geometriemodell . . . . .	57
20	Komplexes Geometriemodell . . . . .	58
21	Die Attributklassen und ihre Beziehungen . . . . .	60
22	Die Struktur einer Komplexobjektart, bestehend aus zwei Objektarten . . .	64
23	Beispiel für ein Straßennetz ohne Komplexobjekte . . . . .	65
24	E/R-Diagramm des DLM-Datenmodells für eine Objektart . . . . .	68
25	Die Tabellen der ATKIS-Datenbank . . . . .	71
26	Beispiel für die Objektbildung in Netzen . . . . .	74
27	Beispiel für mehrfache Referenzen . . . . .	76
28	Modellierung von Brücken . . . . .	77

# Abkürzungsverzeichnis

1NF	Erste Normalform
AA Kart	AdV–Arbeitskreis Kartographie
AAT	Arc Attribute Table
AdV	Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland
ALB	Automatisiertes Liegenschaftsbuch
ALK	Automatisierte Liegenschaftskarte der Landesvermessungs- anstalten
ALK– GIAP	Programmsystem Graphisch– Interaktiver Arbeitsplatz im Rahmen der AdV–Vorhaben ALK und ATKIS
AML	ARC Macro Language
ARC/INFO	Bezeichnung des GIS ARC und des relationalen Datenbanksystems INFO von ESRI
ATKIS	Amtliches Topographisch–Kartographisches Informationssystem
ATT	Attribut gemäß OK
BODIS	Bodeninformationssystem
BZSN	Bezieher Sekundärnachweis
CAD	Computer Aided Design
DBA	Database Administrator
DBI	Database Integrator
DBMS	Datenbankmanagementsystem
DDL	Data Definition Language
DGK	Deutsche Grundkarte
DHM	Digitales Höhenmodell
DLM	Digitales Landschaftsmodell
DKM	Digitales Kartographisches Modell
DSM	Digitales Situationsmodell
DST	Datenstrukturtabelle
DTK	Digitale topographische Karte
DXF	Data Exchange Format
E/R	Entity/Relationship
EDBS	Einheitliche Datenbankschnittstelle
EURODB	Digital Cartographic Data Base of Europe
FAT	Feature Attribut Tabelle
GEWIS	Gewässerinformationssystem

GDB	Graphische Datenbank
GF	Grundfläche
GIS	Geo-Informationssystem
HUIS	Hamburger Umweltinformationssystem
IfAG	Institut für Angewandte Geodäsie
IWK	Internationale Weltkarte
KOAR	Komplexobjektart
KPS	Kartographisches Programmsystem des IfaG
MEGRIN	Multipurpos European Ground Related Information Network
MERKIS	Maßstabsorientiertes Einheitliches Raumbezugssystem für Kommunale Informationssysteme
MF	Maschenfläche
NAT	Node Attribute Table
NF <sup>2</sup>	Non first normal form (nicht erste Normalform)
OAR	Objektart
OJ	ATKIS-Objekt
OK	Objektartenkatalog
OT	ATKIS-Objektteil
PAT	Polygon Attribute Table
RAT	Route Attribute Table
RDBMS	Relationales DBMS
SK	Signaturenkatalog
SQL	Structured Query Language
STABIS	Statistisches Bodeninformationssystem
TK	Topographische Karte
TOPIS	Topographisches Informationssystem der Bundeswehr
TÜK	Topographische Übersichtskarte
ÜF	Überlagerungsfläche
ÜK	Übersichtskarte



# Glossar

**Ableitbare Geschichte:** Geschichte von Objektzuständen, die sich sprunghaft und zwischen diesen Zeitpunkten kontinuierlich verändern. Die kontinuierliche Änderung folgt einem bekannten funktionalen Zusammenhang.

**Arbeitsversion (working version):** Eine Arbeitsversion eines Objektes kann gelöscht, aber nicht geändert werden. Sie besitzt eine mittlere Robustheit. (vgl. *flüchtige Version* und *freigegebene Version*)

**Attribut:** Ein Attribut bezeichnet quantitative und qualitative Eigenschaften, die ein *Objekt* oder ein *Objektteil* näher beschreiben. [ATKIS] Attribute können konkrete Angaben wie Namen und Maße, aber auch durch vierstellige Zahlen kodierte Angaben enthalten. Sie werden als Unterscheidungskriterium von *Objekten* und *Objektteilen* herangezogen.

**Attribut, DLM-abhängiges:** Bei der Abbildung eines realen Landschaftsobjektes in verschiedenen DLM kann dieses Attribut unterschiedlich belegt sein.

**Attribut, DLM-übergreifendes:** Bei der Abbildung eines realen Landschaftsobjektes ist die Belegung dieses Attributes unabhängig vom DLM immer gleich.

**Attribut, objektabhängiges:** Eventuelle Übereinstimmungen dieses Attributes bei mehreren Objekten basieren nicht auf semantischen Abhängigkeiten und sollen nicht als solche verwaltet werden.

**Attribut, objektübergreifendes:** Attribute, die aufgrund semantischer Abhängigkeiten für mehrere Objekte identisch sind.

**Attribut, hervorgehobenes:**  $\rightarrow$  *Objektattribut*

**Attribut, klassifiziertes:** Attribut, das mit einem numerischen Kode belegt wird, der für eine Klasse von numerischen Werten steht. Die Definition der Klassen ist dem OK zu entnehmen.

**Attribut, kodiertes:** Attribut, das mit einem vierstelligen numerischen Kode belegt wird. Die Bedeutung der Kodes ist im OK aufgelistet.

**Attribut, stetiges:** Attribut, das mit einem echten numerischen Wert belegt wird.

**Aufzeichnungszeit  $t_a$ :** Zeitpunkt, zu dem eine Aussage in ein Datenbanksystem eingebracht wurde.

**Aufzeichnungszeitmodell:** Temporales Datenmodell auf der Basis der Aufzeichnungszeit.

Es wird Aufzeichnungszeit gleich Gültigkeitszeit bzw. Beginn des Gültigkeitszeitintervalls angenommen.

**Basisflächen:** entsprechen  $\rightarrow$  *Maschenflächen*, wurden zu besserer Unterscheidung von den *Grundflächen* umbenannt

**Beförderung (promotion) einer Version:** Vorgang der Überführung einer *flüchtigen Version* in eine *Arbeitsversion* und von einer *Arbeitsversion* in eine *freigegebene Version*.

**Beobachtungszeit  $t_b$ :** Zeitpunkt, zu dem die Richtigkeit einer Aussage erkannt wurde.

**Bitemporales Modell:** Temporales Datenmodell auf der Basis der von einander unabhängigen Aufzeichnungs- und Gültigkeitszeit. Die Gültigkeitszeit kann ein Zeitpunkt oder ein Zeitintervall sein.

**Chronon:** In Bezug auf das Maßsystem kleinste sinnvoll zu unterscheidende Zeiteinheit.

**Coverage:** Auch *Cover*. Mechanismus von ARC/INFO zur Verwaltung und Speicherung einer räumlich abgegrenzten Menge von *spatialen* und damit verbundenen *aspatialen* Daten. Physisch wird jedes Coverage in einem eigenen Verzeichnis gespeichert.

**Daten, aspatiale:** Alle Daten, die nicht den *spatialen* Daten zuzurechnen sind.

**Daten, spatiale:** Geometrische Daten einschließlich komplexer Geometriedaten, die selbst keinen eigentlichen Raumbezug mehr besitzen.

**Degradierung (demote) einer Version:** Vorgang der Überführung einer *Arbeitsversion* in eine *flüchtige Version*. Eine *freigegebene Version* kann nicht degradiert werden.

**Entity/Relationship-modell:**

**Ereignisorientierte Geschichte:** Geschichte von Objektzuständen, die nur zu gewissen Zeitpunkten definiert sind. Es kann Zeitpunkte ohne und Zeitpunkte mit mehreren Objektzuständen geben.

**Flüchtige Version (transient version):** Eine flüchtige Version eines Objektes ist eine temporäre Version, die geändert und gelöscht werden kann. Für das Objekt geforderten Integritätsbedingungen können bei flüchtigen Versionen für ungültig erklärt werden. Sie besitzt die geringste Robustheit. (vgl. *Arbeitsversion* und *freigegebene Version*)

**Freigegebene Version (released version)** : Eine freigegebene Version eines Objektes kann weder geändert noch gelöscht werden. Sie besitzt die höchste Robustheit. (vgl. *flüchtige Version* und *Arbeitsversion*)

**Geo-Informationssystem:** Ein Geo-Informationssystem ist ein Informationssystem, das der Bereitstellung von Fachinformationen unter Berücksichtigung ihres Raumbezuges dient. Es bietet Funktionen zur Erfassung, Bearbeitung und Darstellung von raumbezogenen Daten an. Es integriert geometrische Primitive, graphische und thematische Beschreibungen zu raumbezogenen Objekten. [Pag93]

Ein GIS ist ein DBMS, mit dem spatiale und aspatiale Daten erfaßt, gespeichert, verwaltet, ausgewertet, dargestellt und in Beziehung zueinander gebracht werden können.

**Geschichte:** Eine Geschichte ist eine zeitlich geordnete Folge von Zuständen oder Ereignissen von Aussagen, die zu einem Objekt oder einer Beziehung gehören. [Här84]

**Granularität, konstante:** Eine Granularität ist eine Einteilung einer linearen Zeitachse oder eines Ausschnittes einer linearen Zeitachse und wird bestimmt durch eine Länge und einen Ausgangspunkt. Die Länge wie auch der Ausgangspunkt werden in Chronons der zugrundeliegenden Uhr angegeben.

**Granularität, variable:** Eine Granularität, deren Länge im Gegensatz zur *konstanten Granularität* variabel sein kann.

**Granulat:** Bezugsmaß für die durch die Granularität definierten Zeitabschnitte. Ein Granulat besitzt eine konstante Größe bei konstanter Granularität der Zeit.

**Grundflächen:** Grundflächen dürfen sich nur in Ausnahmefällen gegenseitig überlagern. In diesem Fall ist eine der Grundflächen wie eine *Überlagerungsfläche* zu behandeln. Sie können von einer oder mehreren *Überlagerungsflächen* überlagert werden. Es gibt Objektarten, welche sowohl Überlagerungs-, als auch Grundflächen sein können. Allerdings können auch diese OAR im konkreten Fall nur einen der beiden Zustände annehmen.

**Gültigkeitszeit  $t_g$ :** Zeitpunkt oder Zeitintervall, in dem eine Aussage wahr ist bzw. als wahr angesehen wird. Die Gültigkeitszeit kann in eine oder beide Richtungen unbeschränkt sein.

**Gültigkeitszeitmodell:** Temporales Datenmodell auf der Basis der Gültigkeitszeit. Aufzeichnungszeit und Beobachtungszeit werden nicht beachtet.

**Netz, topologisches:** Komplexe netzartige Struktur, bestehend aus thematisch eng verwandten, räumlich miteinander verbundenen Landschaftsobjekten, meist mit ähnlicher geometrischer Ausprägung. Bedeutende Vertreten sind das Straßen-, Eisenbahn- und Gewässernetz (siehe Tabelle 12 auf Seite 75).

**Objekt, topographisches:** Konkreter, einer *Objektart* zugeordneter, geometrisch begrenzter, durch *Attribute* beschriebener und mit Namen versehbarer Gegenstand der Landschaft. Es besteht aus einer nach fachlichen Gesichtspunkten gebildeten Menge von *Objektteilen*. Topographische Objekte können unter Berücksichtigung des unterschiedlichen Strukturierungsgrades punktförmig, linienförmig oder flächenförmig sein. Auf der Erdoberfläche selbst sind sie stets flächenförmig. [ATKIS]

**Objektart:** Zusammenfassende Bezeichnung für gleichartige *topographische Objekte* zum Zwecke der Katalogisierung. [ATKIS]

**Objektattribut:** Attribut, das der Objektebene zugeordnet ist.

**Objektbereich:** Höchste Stufe in der ATKIS-Begriffsorganisation, besteht aus *Objektgruppen*. [ATKIS]

**Objektgruppe:** Unter dem *Objektbereich* liegende Begriffseinheit, besteht aus *Objektarten*. [ATKIS]

**Objektteil:** Ein Objektteil ist ein konkreter, geometrisch begrenzter und durch einheitliche *Attribute* und *Relationen* gekennzeichnete Gegenstand der Landschaft als Teil eines *Objekts*. Beim Wechsel eines *Attributs* entsteht ein neuer Objektteil. [ATKIS]

**Objektteilattribut:** Attribut, das der Objektteilebene zugeordnet ist.

**Objekttyp:** Legt fest, ob ein Objekt punkt-, linien- oder flächenförmig zu modellieren ist.

**Landschaftsobjekt:** → *Objekt, topographisches*

**Masche:** Durch die topologischen Netze abgegrenzte zusammenhängende Fläche der Landschaft. Eine Masche kann hineinragende Objektteile beinhalten. Alle durch die Netze entstehenden Maschen werden mit einem oder mehreren Objekten der zu den *Maschenflächen* gerechneten Objektarten gefüllt, d.h. diese OAR dürfen keine Objekte der Netze überlappen. Eine vollständige Füllung aller Maschen ist jedoch nur im DLM 25 vorgesehen.

**Modellgenauigkeit:** Maß für die Lagegenauigkeit und die Mindestgröße von Objekten. Unterschreitet ein Landschaftsobjekt die Mindestgröße, so wird es nur in Ausnahmefällen erfaßt und in das DLM übernommen. Die Modellgenauigkeit ist abhängig

vom DLM und entspricht i.A. den Bestimmungen der zugrundegelegten Topographischen Kartenwerke des entsprechenden Maßstabes.

**Modellobjekt:** Die Abbildung eines topographischen Objektes in der DLM-Datenbank. Auch als DLM-Objekt bezeichnet.

**Modelltyp:**  $\rightarrow$  *Objekttyp*

**Todeszeit:** Zeitpunkt, ab dem ein Objekt keine gültige Version mehr besitzt.

**Topologie:** Daten über die dreidimensionale Struktur der Erdoberfläche.

**überlagern:** Zwei Objekte oder Objektteile überlagern sich, wenn sie in Bezug auf ihre geometrische Repräsentation eine gemeinsame Teilfläche bzw. bei linienförmiger Modellierung eine gemeinsame Linie besitzen.

**Überlagerungsflächen:** Flächenobjekte, die nur kombiniert mit einem Grundflächenobjekt auftreten. Dabei sind Grund- und Überlagerungsfläche zumindest teilweise derselben Geometrie zugeordnet. Zu einer konkreten Grundfläche können mehrere Überlagerungsflächen existieren. Es gibt Objektarten, welche sowohl Überlagerungs-, als auch Grundflächen sein können. Allerdings können auch diese OAR im konkreten Fall nur einen der beiden Zustände annehmen.

**Version:** Eine Version ist ein Zustand eines Objektes. Sie besitzt einen Gültigkeitszeitraum und ist je nach Art der Version unveränderlich oder veränderlich. Nach der Erzeugung einer ersten Version werden alle weiteren Versionen eines Objektes von dieser rekursiv abgeleitet und bilden eine Versionenhierarchie.

**Zeitdauer:** Die Zeitdauer eines Zeitintervalls im technischen Sinne ist die Differenz von Anfangs- und Endpunkt des Intervalls in Chronons. Dabei wird für den Anfangspunkt der Beginn und für den Endpunkt das Ende des zugeordneten Zeitabschnittes auf dem Zeitstrahl gerechnet. Bei konstanter Granularität kann die Zeitdauer auch in Granulaten gerechnet werden.

**Zeitintervall:** Abbildung einer *natürlichen* Zeitspanne auf die betrachtete Granularisierung der Zeit. Dabei wird der Anfangs- und Endpunkt der Zeitspanne auf die entsprechenden *technischen* Zeitpunkte abgebildet.

**Zeitpunkt, technischer:** Im Unterschied zum eigentlichen Sinne des Wortes beschreibt ein technischer Zeitpunkt nicht einen konkreten Punkt, sondern ein Abschnitt der Zeitachse entsprechend der betrachteten Granularität. Dieser *technische* Zeitpunkt

wird aus einem *natürlichen* Zeitpunkt durch Abbildung auf den entsprechenden Abschnitt der betrachteten Granularität gewonnen.

**Zeitstempel:** Die Kombination von Zeitpunkten und –intervallen, die für jede Version eines Objektes gespeichert werden.

**Zustandserhaltende Geschichte:** Geschichte von Aussagen, die jeweils einen neuen Zustand eines Objektes oder einer Beziehung beschreibt. Ein Zustand bleibt bis zum Inkrafttreten eines neuen Zustandes invariant.

**Zustandsverändernde Geschichte:** Geschichte von kontinuierlich veränderlichen Objektzuständen. Der funktionale Zusammenhang von Zeit und Zustand ist stetig, aber unbekannt. Die Erfassung kann exemplarisch durch Zustände hinreichender zeitlicher Dichte erfolgen.

# Literatur

- [AGH96] AG Hannover im Vorhaben ALK/ATKIS: *Bezieher Sekundärnachweis*. Niedersächsisches Landesverwaltungsamt, Landesvermessung, Okt. 1996
- [AKK93] Expertengruppe DKM/SK der AdV: *Bericht zur 56. AKKart-Tagung, 24. und 25. Mai 1993 in Bad Kühlungsborn*. AdV, 1993
- [ARC] *ARC/INFO Data Management*. Environmental Systems Research Institute Inc., New York, 1994
- [ATKIS] AdV: *Amtliches Topographisch-Kartographisches Informationssystem ATKIS, Gesamtdokumentation*. AdV, Hannover, 1995
- [Bar95] Norbert Bartelme: *Geoinformatik*. Springer-Verlag Berlin, Heidelberg, 1995
- [Ben79] J. Ben-Zvi et al.: *PGS-1, Payroll Generator System-1*. In Proceedings of the I.P.A. Conference, S.309-332, Tel-Aviv, Israel, 1979
- [Ben82] J. Ben-Zvi: *The Time Relational Model*. PhD. Diss., Computer Science Department, UCLA, 1982
- [Ber] Jochen van den Bercken, Bernhard Seeger: *Query Processing Techniques for Multiversion Access Methods*. Universität Marburg, technischer Report
- [Cat91] Roderic Geoffrey Galton Cattell: *Object Data Management*. Addison-Wesley Publishing Company, 1991
- [Chr92] Friedrich Christoffers: *Rahmenbedingungen zur Einrichtung des ATKIS-DLM 25/1 in Niedersachsen*. Nachrichten der Niedersächsischen Vermessungs- und Katasterverwaltung, 42. Jahrg., Heft 3, S. 121 - 133, 1992
- [Cli83] James Clifford, D. S. Warren: *Formal Semantics for Time in Databases*. ACM TODS, Vol. 8, No. 2, S. 214 - 254, Juni 1983
- [Cli86] J. Clifford, G. Ariav: *Temporal Data Management: Models and Systems*. Ablex Publishing Corporation, Norwood, New Jersey, 1986
- [End93] Manfred Endrullis: *Digitale Kartographie im IfAG*. Kartographische Schriften, Band 1, S.81 - 87, Kirschbaum Verlag GmbH, Bonn, September 1993
- [End95] Manfred Endrullis, Achim Hoppe, Kerstin Reinhold, Ingo Wilski: *Sammlung und Zusammenführung von Verwaltungsgrenzen-Datensätzen der Bundesrepublik Deutschland*. Kartographische Nachrichten, 45. Jahrg., Heft 5, S. 165 - 172, Kirschbaum Verlag GmbH, Bonn, Oktober 1995
- [GIS] *Understanding GIS*. Environmental Systems Research Institute Inc., New York, 1994
- [Gri95/1] Bernd Griefahn: *Zugriffsmodelle für temporale Daten*. Rostocker Informatik-Berichte 17, 1995

- [Gri95/2] Bernd Griefahn: *Eine Erweiterung der relationalen Algebra für temporale Attribute mit unterschiedlichen Zeitfolgetypen*. Universität Rostock, technischer Report, 1995
- [Gri96] Bernd Griefahn: *Implementierungstechniken für temporale Daten*. Universität Rostock, technischer Report, 1996
- [Här84] Th. Härder: *Überlegungen zur Modellierung und Integration der Zeit in temporalen Datenbanksystemen*. Bericht Nr. 19/84, Universität Kaiserslautern, Okt. 1984
- [Här86] Th. Härder: *Datenbanksysteme I*. Skriptum zur Vorlesung, Universität Kaiserslautern, 1986
- [Har93] Rolf Harbeck: *Basisinformationssysteme des Vermessungswesens*. Kartographische Schriften, Band 1, S. 60 – 68, Kirschbaum Verlag GmbH, Bonn, September 1993
- [Hug92] John G. Hughes: *Objektorientierte Datenbanken*. Hanser, Prentice-Hall, München, Wien, London, 1992
- [Jäg93] Ernst Jäger: *Vom digitalen kartographischen Modell zur Karte*. Kartographische Schriften, Band 1, S. 69 – 80, Kirschbaum Verlag GmbH, Bonn, September 1993
- [Jen92] Christian S. Jensen, J. Clifford, S. K. Gadia, A. Segev, R. T. Snodgrass: *A glossary of temporal database concepts*. SIGMOD RECORD, 21(3):35–43, Sep. 1992
- [Jen93/1] Christian S. Jensen, J. Clifford, R. Elmasri, S. K. Gadia, P. Hayes, S. Jajodia: *A Consensus Glossary of Temporal Database Concepts*. Aalborg University, Technical Report R 93–2035, November 1993
- [Jen93/2] Christian S. Jensen, Richard T. Snodgrass: *The TEMPIS Project. Proposal for a Data Model for the Temporal Structured Query Language*. TEMPIS Technical Report No. 37, University of Arizona, Tucson, AZ 85721, 1993
- [Käf92] Wolfgang Käfer: *Geschichts- und Versionenmodellierung*. Dissertation, Universität Karlsruhe, 1992
- [Kim90] Won Kim: *Introduction to Object-Oriented Databases*. The MIT Press, Massachusetts Institute of Technology, Cambridge, 1990
- [Kim95] Won Kim: *Modern Database Systems*. ACM Press, 1995
- [Kop90] Helmut Kopka: *L<sup>A</sup>T<sub>E</sub>X – Erweiterungsmöglichkeiten*. Addison-Wesley Co., Bonn, 1990
- [Kop94] Helmut Kopka: *L<sup>A</sup>T<sub>E</sub>X – Einführung*. Addison-Wesley Co., Bonn, 1994
- [Lan95] Arne Lange: *Vergleich SQL-artiger Anfragesprachen Temporaler Datenbanksysteme*. Universität Rostock, Studienarbeit, 1995
- [Lan96] Arne Lange: *Anbindung der temporalen Anfragesprache TSQL2 an ein objektorientiertes Datenbanksystem*. Universität Rostock, Diplomarbeit, 1996



- [Lud95] Monika Ludwig: *Temporale Datenmodelle – Ein vergleichender Überblick*. Universität Rostock, Studienarbeit, 1995
- [Lud96] Monika Ludwig: *Realisierungskonzepte für temporale Attribute*. Universität Rostock, Diplomarbeit, 1996
- [Med] Claudia Bauzer Medeiros, Fatima Pires: *Databases for GIS*.
- [Org93] Mehmet A. Orgun, Hausi A. Müller: *A Temporal Algebra Based on an Abstract*. In Proceedings of the ADC'93, 1993
- [Pag93] Page, Häuslein, Greve: *Das Hamburger Umweltinformationssystem HUIS*. S.83 ff, 1993
- [Pet94] Dušan Petković: *Sybase- und Microsoft-SQL Server*. Addison-Wesley Co., Bonn, 1994
- [Pis94] Niki Pissinou u.a.: *Towards an Infrastructure for Temporal Databases: Report of an Invitational ARPA/NSF Workshop*. University of Arizona, Technical Report 94-01, Januar 1994
- [Sam95/1] Hanan Samet, Walid G. Aref: *Spatial Data Models and Query Processing*. Modern Database Systems, S. 338 – 360, ACM Press, 1995
- [Sam95/2] Hanan Samet: *Spatial Data Structures*. Modern Database Systems, S. 361 – 385, ACM Press, 1995
- [Sno90] R. Snodgrass: *Temporal databases: Status and research directions*. ACM SIGMOD Record, 19(4):83–89, Dez. 1990
- [Sno93] Richard T. Snodgrass: *Overview of the Special Section on Temporal Database Infrastructure*. University of Arizona, 1993
- [Sno95/1] Richard T. Snodgrass, The TSQL2 Language Design Committee: *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, Boston, Dordrecht, London, 1995
- [Sno95/2] Richard T. Snodgrass: *Temporal Object-Oriented Databases: A Critical Comparison*. Modern Database Systems, S. 386 – 408, ACM Press, 1995
- [Tan93] Abdullah Uz Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, R. T. Snodgrass: *Temporal databases: theory, design, and implementation*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, 1993
- [Wed] Hartmut Wedekind: *Are the Terms „Version“ and „Variant“ Orthogonal to One Another*. Universität Erlangen–Nürnberg

# Literatur zu temporalen Datenmodellen

- [1] M. Abadi, Z. Manna: *Temporal logic programming*. In Proceedings of 1987 ISSS Symposium on Logic Programming, San Francisco, CA, Aug. 1987
- [2] M. Abadi, Z. Manna: *Temporal logic programming*. Journal of Symbolic Computation, 8:277-295, 1989
- [3] M. E. Adiba, N. Bui Quang: *Historical multi-media databases*. In Proceeding of the Conference on Very Large Databases. S.63-70, Kyoto, Japan, Aug. 1986
- [4] I. Ahn: *Performance modeling and access methods for temporal database management systems*. Ph. D. thesis, Computer Science Department, University of North Carolina at Chapel Hill, Juli 1986
- [5] I. Ahn: *Towards an implementation of database management systems with temporal support*. In Proceeding of the international Conference on Data Engineering, S.374-381, Los Angeles, CA, Feb. 1986
- [6] I. Ahn, R. Snodgrass: *Performance evaluation of a temporal database management system*. In Proceedings of ACM SIGMOD International Conference on Management of Data, Association for Computer Machinery, S.96-107, Washington, DC, Mai 1986
- [7] I. Ahn, R. Snodgrass: *Partitioned storage for temporal databases*. Information Systems, 13(4):369-391, 1988
- [8] J. F. Allen, P. J. Hayes: *A common-sense theory of time*. In Proceeding of the International Joint Conference on Artificial Intelligence, S.528-531, Los Angeles, CA, Aug. 1985
- [9] J. F. Allen: *Maintaining knowledge about temporal intervals*. Communications of the ACM, 26(11):832-843, Nov. 1983
- [10] J. F. Allen: *Planning as temporal reasoning*. In Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, MA, 1991
- [11] T. L. Anderson: *Modeling time at the conceptual level*. In Proceedings of the International Conference on Databases: Improving Usability and Responsiveness, Academic Press, S.273-297, Jerusalem, Israel, Juni 1982,
- [12] G. Ariav, A. Beller, H. L. Morgan: *A temporal data model*. Technical Report DS-WP 82-12-05, Decision Sciences Department, University of Pennsylvania, Dez. 1994
- [13] G. Ariav, J. Clifford: *Temporal Data Management: Models and Systems*. Ablex Publishing Corporation, Norwood, New Jersey, 1986
- [14] G. Ariav: *A temporally oriented data model*. ACM Transactions on Database Systems, 11(4):499-527, Dez. 1986
- [15] G. Ariav, H. L. Morgan: *MDM: Embedding the time dimension in information systems*. Technical Report 82-03-01, Department of Decision Sciences, The Wharton School, University of Pennsylvania, 1982
- [16] G. Ariav: *Tools for managing temporally oriented data, are they really practically relevant*. In Proceedings of the International Workshop on an Infrastructure for Temporal Databases, Arlington, TX, Juni 1993
- [17] J. Banerjee, W. Kim, H. J. Kim, H. F. Korth: *Semantics and implementation of schema evolution in object-oriented databases*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.311-322, San Francisco, CA, 1987
- [18] R. Barrera, A. Frank, K. Al-Taha: *Temporal Relations in Geographic Information Systems: A Workshop at the University of Maine*. SIGMOD Record, 1991, 20:85-91
- [19] M. Baudinet, M. Niézette, P. Wolper: *On the representation of infinite temporal data and queries (extended abstract)*. In Proceedings of the ACM Symposium on Principles of Database Systems, Association for Computing Machinery, S.280-290, Denver, CO, Mai 1991
- [20] D. Beech, B. Mahbod: *Generalized Version Control in an Object-Oriented Database*. In Proceedings of the International Conference on Data Engineering, S.14-22, 1988
- [21] B. Becker, S. Gschwind, T. Ohler, B. Seeger, P. Widmayer: *On Optimal Multiversion Access Structures*. In Proceedings of the Symposium on Large Spatial Databases, in Lecture Note in Computer Science, Vol. 692, S.123-141, Singapur, 1993
- [22] J. Ben-Zvi et al.: *PGS-1, Payroll Generator System-1*. In Proceedings of the I.P.A. Conference, S.309-332, Tel-Aviv, Israel, 1979
- [23] J. Ben-Zvi: *The Time Relational Model*. PhD. Diss., Computer Science Department, UCLA, 1982
- [24] J. Ben-Zvi: *A time-view of data in a relational database system*. Master's thesis, Computer Science Department, UCLA, Juni 1981
- [25] Jochen van den Bercken, Bernhard Seeger: *Query Processing Techniques for Multiversion Access Methods*. Universität Marburg, technischer Report
- [26] G. Bhargava, S. K. Gadia: *A 2-dimensional temporal relational database model for querying errors and updates, and for achieving zero information-loss*. Technical Report 89-24, Department of Computer Science, Iowa State University, Ames, Iowa, Dez. 1989
- [27] G. Bhargava, S. K. Gadia: *The concept of error in a database: An application of temporal databases*. In Proceedings of 1990 COMAD International Conference on Management of Data, Tata McGraw-Hill, S.106-121, New Delhi, Dez 1990
- [28] R. L. Blum: *Discovery, confirmation, and incorporation of casual relationships from a large time-oriented clinical data base: The RX project*. Computers and Biomedical Research, 15(2):164-187, 1982
- [29] M. H. Böhlen: *Temporal Database System Implementations*. SIGMOD Record, Vol. 24(4), Dez. 1995
- [30] A. Bolour, T. L. Anderson, L. J. Dekeyser, H. K. T. Wong: *The role of time in information processing: A survey*. SigArt Newsletter, 80:28-48, April 1982
- [31] J. A. Bubenko Jr.: *The Temporal Dimension in Information Modeling*. S.93-118, Northholland, The Netherlands, 1977

- [32] A. P. Buchmann, H. Branding: *On Combining Temporal and Real-Time Databases*. In Proceedings of the International Workshop on an Infrastructure for Temporal Databases, Arlington, TX, Juni 1993
- [33] Roderic Geoffrey Galton Cattell: *Object Data Management*. Addison-Wesley Publishing Company, 1991
- [34] W. Cellary, G. Jomier: *Consistency of versions in object-oriented databases*. In Proceedings of the Conference on Very Large Databases, Brisbane, Australia, Aug. 1990
- [35] S. Chaudhuri: *Temporal relationships in databases*. In Proceedings of the Conference on Very Large Databases, Los Angeles, CA, Aug. 1988
- [36] J. Chomicki, T. Imielinski: *Temporal deductive databases and infinite objects*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.174-183, Mai 1989
- [37] J. Chomicki: *Polynomial time computable queries in temporal deductive databases*. In Ninth Annual ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Nashville, TN, April 1990
- [38] E. Ciapessoni, E. Corsetti, A. Montanari, P. San Pietro: *Embedding time granularity in a logical specification language for synchronous real-time systems*. Science of Computer Programming, Jan. 1992, Überarbeitung Mai 1992
- [39] J. Clifford: *A logical framework for the temporal semantics and natural-language querying of historical databases*. Ph. D. Diss., Department of Computer Science, State University of New York at Stonybrook, Dez. 1982
- [40] J. Clifford: *A model for historical databases*. In Proceedings of Workshop on Logical Bases for Data Bases, Toulouse, Frankreich, Dez. 1982
- [41] J. Clifford, D. S. Warren: *Formal Semantics for Time in Databases*. ACM Transactions on Database Systems (TODS), Vol. 8, No. 2, S. 214 - 254, Juni 1983
- [42] J. Clifford, A. U. Tansel: *On an algebra for historical relational databases: Two views*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.247-265, Austin, TX, Mai 1985
- [43] J. Clifford, A. Croker: *The historical relational data model (HRDM) and algebra based on lifespans*. In Proceedings of the Third International Conference on Data Engineering, S.528-537, Los Angeles, CA, Feb 1987
- [44] J. Clifford, A. Rao: *A simple, general structure for temporal domains*. In Proceedings of the Conference on Temporal Aspects in Information Systems, AFCET, S.23-30, Frankreich, Mai 1987
- [45] J. Clifford, A. Croker: *Objects in time*. IEEE Data Engineering, 7(4):189-196, Dez. 1988
- [46] J. Clifford, A. Croker, A. Tuzhilin: *On completeness of historical relational query languages*. Technical Report IS-91-41, ACM Transactions on Database Systems, New York University, Dez. 1991
- [47] J. Clifford, A. Croker, A. Tuzhilin: *On completeness of historical relational query languages*. Technical Report STERN IS-93-8, New York University Stern School of Business, 1993
- [48] E. Corsetti, A. Montanari, E. Ratto: *Dealing with different time granularities in formal specifications of real-time systems*. Journal of Real-Time Systems, III, Juni 1991
- [49] P. Dadam, V. Lum, H. D. Werner: *Integration of time versions into a relational database systems*. In Proceedings of the Conference on Very Large Databases, S.509-522, Singapore, Aug. 1984
- [50] P. Dadam, J. Teuhola: *Managing Schema Version in a Time-versioned Non-First-Normal-Form Relational Database*. In Proceedings of BTW, S.161-179, Darmstadt, 1987
- [51] L. Dami, E. Fiume, O. Nierstrasz, D. Tschritzis: *Temporal scripts for objects*. Technical Report, Centre Universitaire d'Informatique, University of Geneva, Schweiz, 1988
- [52] C. J. Date: *A proposal for adding date and time support to SQL*. SIGMOD Record, 17(2):53-76, Juni 1988
- [53] U. Dayal, B. Blaustein, A. Buchmann, U. Chakravarthy, M. Hsu, R. Ladin, D. R. McCarthy, A. Rosenthal, S. Sarin, M. J. Carey, M. Livny, R. Jauhari: *The HIPAC project: Combining active database and timing constraints*. ACM SIGMOD Record, 17(1):51-70, März 1988
- [54] U. Dayal, G. Wu: *A uniform approach to processing temporal queries*. In Proceedings of the Conference on Very Large Databases, Vancouver, Canada, Aug. 1992
- [55] T. Dean, D. V. McDermott: *Temporal data base management*. Artificial Intelligence, 1987
- [56] R. Dechter, I. Meiri, J. Pearl: *Temporal constraint networks*. Artificial Intelligence, 49:61-95, 1991
- [57] K. R. Dittrich, R. A. Lorie: *Version Support for Engineering Database Systems*. IEEE Transactions on Software Engineering, 14(4):129-137, 1988
- [58] G. Dong, S. Ginsburg: *Localizable constraints for object histories*. Technical Report TR-86-217, Computer Science Department, University of Southern California, 1986
- [59] S. Dutta: *Generalized events in temporal databases*. In Proceedings of the Fifth International Conference on Data Engineering, S.118-125, Feb. 1989
- [60] C. E. Dyreson, R. T. Snodgrass: *Historical indeterminacy*. Technical Report TR-91-30A, Computer Science Department, University of Arizona, April 1992
- [61] C. E. Dyreson, R. T. Snodgrass: *Time-stamps semantics and representation*. TempIS Technical Report 33, Computer Science Department, University of Arizona, Mai 1992
- [62] I. El-Assab: *Query language constructs for temporal databases*. Master's thesis, Department of Computer Science, University of Houston, Dez. 1990
- [63] R. Elmasri, G. Wu: *A temporal model and query language for ER databases*. In Proceedings of the Sixth International Conference on Data Engineering, S.76-83, Feb. 1990

- [64] R. Elmasri, G. Wu, Y. Kim: *The time index: An access structure for temporal data*. In Proceedings of the Conference on Very Large Databases, Brisbane, Australien, Aug. 1990
- [65] R. Elmasri, I. El-Assal, V. Kouramajian: *Semantics of temporal data in an extended ER model*. In Proceedings of the Ninth International Conference on the Entity-Relationship Approach, Lausanne, Schweiz, Okt. 1990
- [66] O. Etzion, A. Gal, A. Segev: *A temporal active database model*. Technical Report LBL-32587, Lawrence Berkeley Laboratory, 1992
- [67] O. Etzion, A. Gal, A. Segev: *Temporal Support in Active Databases*. In Proceedings of the Second Workshop on Information Technologies and Systems (WITS), Dez. 1992
- [68] O. Etzion, A. Gal, A. Segev: *Temporal active databases*. In Proceedings of the International Workshop on an Infrastructure for Temporal Databases, Juni 1993
- [69] S. Ferg.: *Modeling the time dimension in an entity-relationship diagram*. In Proceedings of the Fourth International Conference on the Entity-Relationship Approach, IEEE Computer Society Press, S.280-286, Silver Spring, 1985
- [70] A. U. Frank, I. Campari, U. Formentini: *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. In Proceedings of the International Conference Gis - From Space to Territory, Pisa, Italien, Sep. 1992
- [71] S. K. Gadia, J. H. Vaishnav: *A query language for a homogeneous temporal database*. In Proceedings of the ACM Symposium on Principles of Database Systems, S.51-56, März 1985
- [72] S. K. Gadia: *Temporal element as a primitive for time in temporal databases and its application in query optimization*. An abstract in Proceedings of the 1986 ACM Computer Sciences Conference, 1986
- [73] S. K. Gadia: *Towards a multihomogeneous model for a temporal database*. In Proceedings of the International Conference on Data Engineering, S.390-397, Los Angeles, CA, Feb. 1986
- [74] S. K. Gadia: *Weak temporal relations*. In Proceedings of the ACM Symposium on Principles of Database Systems, ACM SIGACT-SIGMOD, Los Angeles, CA, 1986
- [75] S. K. Gadia, C. S. Yeung: *Temporal query languages and the first normal form*. Technical report, Iowa State University, Ames, IA, Juli 1987
- [76] S. K. Gadia, C. S. Yeung: *A generalized model for a relational temporal database*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.251-259, Chicago, IL, Juni 1988
- [77] S. K. Gadia: *A homogeneous relational model and query languages for temporal databases*. ACM Transactions on Database Systems, 13(4):418-448, Dez. 1988
- [78] S. K. Gadia: *The role of temporal elements in temporal databases*. IEEE Data Engineering, Dez. 1988
- [79] S. K. Gadia, C. S. Yeung: *Inadequacy of interval time stamps in temporal databases*. Information Sciences, 54:1-22, 1991
- [80] S. K. Gadia: *A seamless generic extension of SQL for querying temporal data*. Preliminary paper, Computer Science Department, Iowa State University, März 1992
- [81] S. K. Gadia, V. Chopra, U. S. Tim: *Seamless SQL-like query of spatio-temporal data and a case study in agricultural environment management*. Technical Report TR-92-21, Computer Science Department, Iowa State University, 1992
- [82] S. K. Gadia, S. S. Nair, Y. C. Poon: *Incomplete information in relational temporal databases*. In Proceedings of the 18th International Conference on Very Large Databases, Vancouver, Canada, Aug. 1992
- [83] S. K. Gadia, S. S. Nair, Y. C. Poon: *A relational Model for incomplete information in temporal databases*. Technical Report TR-92-06, Computer Science Department, Iowa State University, 1992
- [84] N. Gehani, H. V. Jagadish, I. S. Mumick, O. Shmueli: *Temporal Queries for Active Database Support*. In Proceedings of the International Workshop on an Infrastructure for Temporal Databases, Arlington, TX, Juni 1993
- [85] S. Ginsburg, K. Tanaka: *Interval queries on object histories: Extended abstract*. In Proceedings of the Conference on Very Large Databases, S.208-217, Singapore, Aug. 1984
- [86] S. Ginsburg, K. Tanaka: *Computation-tuple sequences and object histories*. ACM Transactions on Database Systems, 11(2):186-212, Juni 1986
- [87] S. Ginsburg, C. Tang: *Projection of object histories*. Theoretical Computer Science, (48):297-328, 1986
- [88] S. Ginsburg, M. Gyssens: *Object histories which avoid certain subsequences*. Information and Computation, 73:174-206, 1987
- [89] S. Ginsburg, S. Kurtzman: *Spreadsheet and object-history P-simulation*. Technical Report CRI-87-45, Computer Science Department, University of Southern California, Los Angeles, CA, Nov. 1987
- [90] S. Ginsburg, D. Simovici, X. Wang: *Content-related interval queries on object histories*. Information and Computation, 1989
- [91] S. Ginsburg, C. Tang: *Cohesion of object histories*. Theoretical Computer Science, (63):63-90, 1989
- [92] S. Ginsburg, D. Tian: *Input-dependent-only object histories*. Journal of Computer and System Sciences, (40):346-376, 1990
- [93] I. Goralwalla, A. U. Tansel: *A temporal relational database management systems*. In Proceedings of the Sixth International Symposium on Computer and Information Sciences, S.75-83, Antalya, Türkei, Okt. 1991
- [94] I. Goralwalla, A. U. Tansel: *Performance evaluation of temporal relational databases*. Technical report, Bilkent University, Juli 1992
- [95] I. Goralwalla, A. U. Tansel: *A temporal relational database management system*. Technical report, Bilkent University, Juli 1992

- [96] B. Griefahn: *Zugriffsmodelle für temporale Daten*. Rostocker Informatik-Berichte 17, 1995
- [97] B. Griefahn: *Eine Erweiterung der relationalen Algebra für temporale Attribute mit unterschiedlichen Zeitfolgetypen*. Universität Rostock, technischer Report, 1995
- [98] B. Griefahn: *Zeitdarstellung in temporalen Datenbanken*. Universität Rostock, Fachbereich Informatik, Interner Bericht, Time-Report Nr.8, Rostock, 1995
- [99] B. Griefahn: *Implementierungstechniken für temporale Daten*. Universität Rostock, technischer Report, 1996
- [100] H. Gunadhi, A. Segev, G. Shantikumar: *Selectivity estimation in temporal databases*. Technical Report LBL-27435, Lawrence Berkeley Laboratory, 1989
- [101] H. Gunadhi, A. Segev: *Efficient indexing methods for temporal relations*. Technical Report LBL-28798, University of California, Lawrence Berkeley Laboratory, 1989
- [102] H. Gunadhi, A. Segev: *A framework for query optimization in temporal databases*. In Proceedings of the Fifth International Conference on Statistical and Scientific Database Management Systems, 1989
- [103] H. Gunadhi, A. Segev: *A framework for query optimization in temporal databases*. Technical Report LBL-26417, School of Business Administration and Computer Research, Lawrence Berkeley Laboratory, University of California, Berkeley, CA, 1989
- [104] H. Gunadhi, A. Segev: *A Framework for Query Optimization in Temporal Databases*. Vol.420, S.131-147, Springer Verlag, April 1990
- [105] H. Gunadhi, A. Segev: *Efficient indexing methods for temporal relations*. IEEE Transactions on Knowledge and Data Engineering, 1991
- [106] H. Gunadhi, A. Segev: *Query processing algorithms for temporal intersection joins*. In Proceedings of the Seventh International Conference on Data Engineering, Kobe, Japan, 1991
- [107] Th. Härder: *Überlegungen zur Modellierung und Integration der Zeit in temporalen Datenbanksystemen*. Bericht Nr. 19/84, Universität Kaiserslautern, Okt. 1984
- [108] Th. Härder: *Datenbanksysteme I*. Skriptum zur Vorlesung, Universität Kaiserslautern, 1986
- [109] S. H. Hsu, R. T. Snodgrass: *Optimal block size for repeating attributes*. TempIS Technical Report No. 28, Department of Computer Science, University of Arizona, Dez 1991
- [110] S. H. Hsu: *Page and tuple storage structures for historical databases*. TempIS Technical Report No. 34, Computer Science Department, University of Arizona, Tucson, AZ, Mai 1992
- [111] J. G. Hughes: *Objektorientierte Datenbanken*. Hanser, Prentice-Hall, München, Wien, London, 1992
- [112] C. S. Jensen, L. Mark: *A framework for vacuuming temporal databases*. Technical Report CS-TR-2516/UMIACS-TR-90-105, University of Maryland, Department of Computer Science, College Park, MD, Aug. 1990
- [113] C. S. Jensen, L. Mark, N. Roussopoulos: *Incremental implementation model for relational databases with transaction time*. IEEE Transaction on Knowledge and Data Engineering, 3(4):461-473, Dez. 1991
- [114] C. S. Jensen, L. Mark, N. Roussopoulos, T. K. Sellis: *Using caching, cache indexing, and differential techniques to efficiently support transaction time*. VLDB Journal, 1992
- [115] C. S. Jensen, M. D. Soo, R. T. Snodgrass: *Unification of Temporal Data Models*. Technical Report 92-15, University of Arizona, Tucson, 1992
- [116] C. S. Jensen, R. Snodgrass: *Temporal specialization*. IEEE, In Proceedings of the International Conference on Data Engineering, S.594-603, Tempe, AZ, Feb.1992
- [117] C. S. Jensen, J. Clifford, S. K. Gadia, A. Segev, R. T. Snodgrass: *A glossary of temporal database concepts*. ACM SIGMOD RECORD, 21(3):35-43, Sep. 1992
- [118] C. S. Jensen, L. Mark, N. Roussopoulos, T. K. Sellis: *Using Different Techniques to Efficiently Support Transaction Time*. VLDB Journal, 2(1):75-111, 1993
- [119] C. S. Jensen, R. Snodgrass: *Temporal specialization and generalization*. IEEE Transactions on Knowledge and Data Engineering, 1993
- [120] C. S. Jensen, J. Clifford, R. Elmasri, S. K. Gadia, A. Segev, P. Hayes, S. Jajodia: *A Consensus Glossary of Temporal Database Concepts*. In Proceedings of the ARPA/NSF International Workshop on an Infrastructure for Temporal Databases, Arlington, TX, Juni 1993
- [121] C. S. Jensen, J. Clifford, R. Elmasri, S. K. Gadia, P. Hayes, S. Jajodia: *A Consensus Glossary of Temporal Database Concepts*. Aalborg University, Technical Report R 93-2035, November 1993
- [122] C. S. Jensen, R. T. Snodgrass: *The TEMPIS Project. Proposal for a Data Model for the Temporal Structured Query Language*. TEMPIS Technical Report No. 37, University of Arizona, Tucson, AZ 85721, 1993
- [123] S. Jones, P. J. Mason: *Handling the time dimension in a data base*. In Proceedings of the International Conference on Data Bases, S.65-83, University of Aberdeen, British Computer Society, Heyden, Juli 1980
- [124] F. Kabanza, J. M. Stevenne, P. Wolper: *Handling infinite temporal data*. In Ninth Annual ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, S.392-403, Nashville, TN, April 1990
- [125] W. Käfer: *Ein Modell zur Integration der Zeit in relationale Datenbanksysteme*. SFB 124, Bericht Nr. 27/88, Universität Kaiserslautern, 1988
- [126] W. Käfer, N. Ritter, H. Schöning: *Support for temporal data by complex objects*. In Proceedings of the 16th International Conference on Very Large Data Bases, Brisbane, Australien, Aug. 1990
- [127] W. Käfer: *Geschichts- und Versionenmodellierung*. Dissertation, Universität Karlsruhe, 1992
- [128] W. Käfer, B. Mitschang: *Flexible design management for CAD frameworks: concept implementation and assessment*. Tagungsband der GI-Fachtagung „Datenbanksysteme in Büro, Technik und Wissenschaft“, Springer-Verlag, S.144-163, 1993

- [129] S. N. Kamens, G. Wiederhold: *An implementation of temporal queries for SQL*. Technical Report, Department of Computer Science, Stanford University, 1990
- [130] R. Katz: *Toward a unified framework for version modeling in engineering databases*. ACM Computing Surveys, 22(4):375–108, Dez. 1990
- [131] H. Kautz, P. Ladkin: *Integrating metric and qualitative temporal reasoning*. In Proceedings of the 10th AAAI, Anaheim, 1991
- [132] Won Kim: *Introduction to Object-Oriented Databases*. The MIT Press, Massachusetts Institute of Technology, Cambridge, 1990
- [133] Won Kim: *Modern Database Systems*. ACM Press, 1995
- [134] L. Kissouras: *An implementation of a temporal relational algebra*. Master's thesis, Department of Computer Science, Birkbeck College, University of London, 1987
- [135] N. Kline: *An Update of the Temporal Database Bibliography*. SIGMOD Record, 22(4):66–80, Dez. 1993
- [136] M. R. Klopprogge: *TERM: An approach to include the time dimension in the entity-relationship model*. In Proceedings of the Second International Conference on the Entity Relationship Approach, S.477–512, Washington, DC, Okt. 1981
- [137] M. R. Klopprogge, P. C. Lockemann: *Modelling information preserving databases: Consequences of the concept of time*. In Proceedings of the Conference on Very Large Databases, S.399–416, Florenz, Italien, Okt. 1983
- [138] G. Knolmayer, T. Myrach: *Zur Abbildung zeitbezogener Daten in betrieblichen Informationssystemen*. Wirtschaftsinformatik 38, 1996
- [139] C. P. Kolovson, M. Stonebraker: *Indexing techniques for historical databases*. In Proceedings of the Fifth International Conference on Data Engineering, S.127–137, Los Angeles, CA, Feb. 1989
- [140] C. P. Kolovson, M. Stonebraker: *S-trees: Database indexing techniques for multidimensional interval data*. Technical Report UCB/ERL M90/35, University of California, Berkeley, CA, April 1990
- [141] C. P. Kolovson: *Indexing techniques for multi-dimensional spatial data and historical data in database management systems*. Ph. D. thesis, University of California, Berkeley, CA, Nov. 1990
- [142] C. P. Kolovson: *Indexing techniques for multi-dimensional spatial data and historical data in database management systems*. Electronics Research Laboratory Technical Report UCB/ERL M90/105, University of California, Berkeley, CA, Nov. 1990
- [143] C. P. Kolovson, M. Stonebraker: *Segment indexes: Dynamic indexing techniques for multi-dimensional interval data*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.138–147, Mai 1991
- [144] A. Kumar, V. Tsotras, C. Faloutsos: *Access Methods for Bi-Temporal Databases*. In Proceedings of the Workshop on Temporal Databases, S.235–254, Zürich, 1995
- [145] S. Kurtzman: *Properties of spreadsheet histories*. Ph. D. thesis, University of Southern California, Los Angeles, CA, Aug. 1991
- [146] Arne Lange: *Vergleich SQL-artiger Anfragesprachen Temporaler Datenbanksysteme*. Universität Rostock, Studienarbeit, 1995
- [147] Arne Lange: *Anbindung der temporalen Anfragesprache TSQL2 an ein objektorientiertes Datenbanksystem*. Universität Rostock, Diplomarbeit, 1996
- [148] G. Lelakis: *Experimental extension of SQL for the Management of intervals*. Master's thesis, National Technical University of Athens, Athen, Griechenland, 1990
- [149] T. Y. C. Leung, R. R. Muntz: *Query processing for temporal databases*. In Proceedings of the Sixth International Conference on Data Engineering, Los Angeles, CA, Feb. 1990
- [150] T. Y. C. Leung, R. R. Muntz: *Generalized data stream indexing and temporal query processing*. In Second International Workshop on Research Issues in Data Engineering: Transaction and Query Processing, Feb. 1992
- [151] T. Y. C. Leung, R. R. Muntz: *Temporal query processing and optimization in multiprocessor database machines*. In Proceedings of the Conference on Very Large Databases, Vancouver, Kanada, Aug. 1992
- [152] T. Y. C. Leung: *Query processing and optimization in temporal database systems*. Ph. D. thesis, Department of Computer Science, University of California, Los Angeles, CA, 1992
- [153] D. B. Lomet, B. Salzberg: *Access methods for multiversion data*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.315–324, Juli 1989
- [154] D. B. Lomet, B. Salzberg: *The performance of a multiversion access method*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.353–363, Atlantic City, NJ, Mai 1990
- [155] D. B. Lomet, B. Salzberg: *Media recovery with Time-Split B-trees*. Technical Report NU-CCS-91-16, College of Computer Science, Northeastern University, 1991
- [156] N. A. Lorentzos, R. G. Johnson: *TRA: A model for a temporal relational algebra*. In Proceedings of the Conference on Temporal Aspects in Information Systems, AFCET, S.99–112, Frankreich, Mai 1987
- [157] N. A. Lorentzos, R. G. Johnson: *Extending relational algebra to manipulate temporal data*. Information Systems, 13(3):289–296, 1988
- [158] N. A. Lorentzos, R. G. Johnson: *An extension of the relational model to support generic intervals*. In Extending Data Base Technology 88 Conference, Venice, Italien, 1988
- [159] N. A. Lorentzos, R. G. Johnson: *Requirements specification for a temporal extension to the relational model*. Data Engineering, 11(4):26–33, 1988
- [160] N. A. Lorentzos, V. Kollias: *The handling of depth and time intervals in soil-information systems*. Computers and Geosciences, 15(3):395–401, 1989

- [161] N. A. Lorentzos: *DBMS support for time and totally ordered compound data types*. Information Systems, 1991
- [162] N. A. Lorentzos, A. Poulouvasilis, C. Small: *Update operations for multi-dimensional interval data and their optimization*. Informatics Laboratory, Agricultural University of Athens, 1992
- [163] Monika Ludwig: *Temporale Datenmodelle – Ein vergleichender Überblick*. Universität Rostock, Studienarbeit, 1995
- [164] Monika Ludwig: *Realisierungskonzepte für temporale Attribute*. Universität Rostock, Diplomarbeit, 1996
- [165] V. Lum, P. Dadam, R. Erbe, J. Günauer, P. Pistor, G. Walch, H. Werner, J. Woodfill: *Designing DBMS support for the temporal dimension*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.115–130, Boston, MA, Juli 1984
- [166] R. Maiocchi, B. Pernici, F. Berbic: *Automatic deduction of temporal informations*. ACM Transactions on Database Systems, 1992
- [167] N. G. Martin, S. B. Navathe, R. Ahmed: *Dealing with temporal schema anomalies in history databases*. In Proceedings of the Conference on Very Large Databases, S.177–184, Brighton, England, Sep. 1987
- [168] E. McKenzie: *Bibliography: Temporal databases*. Proceedings of ACM SIGMOD International Conference on Management of Data, 15(4):40–52, Dez. 1986
- [169] E. McKenzie, R. Snodgrass: *Extending the relational algebra to support transaction time*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.467–478, San Francisco, CA, Mai 1987
- [170] E. McKenzie, R. Snodgrass: *Supporting valid time: An historical algebra*. Technical Report TR87–008, Computer Science Department, University of North Carolina at Chapel Hill, Aug. 1987
- [171] E. McKenzie: *An algebraic language for query and update of temporal databases*. Ph. D. thesis, Computer Science Department, University of North Carolina at Chapel Hill, Sep. 1988
- [172] E. McKenzie, R. Snodgrass: *Schema evolution and the relational algebra*. Information Systems, 15(2):207–232, Juli 1990
- [173] E. McKenzie, R. Snodgrass: *An evaluation of relational algebras incorporating the time dimension in databases*. ACM Computing Surveys, 23(4):501–543, Dez. 1991
- [174] E. McKenzie, R. Snodgrass: *Supporting valid time in an historical relational algebra: Proofs and extensions*. Technical Report TR–91–15, Department of Computer Science, University of Arizona, Tucson, AZ, Aug. 1991
- [175] T. Müller, D. Steinbauer, H. Wedekind: *Control of Versions in Database Applications*. In Proceedings of the IEEE Conference on Trends and Applications, Gaithersburg, MD, S.308–316, 1981
- [176] P. Muth, A. Kreiß, G. Weikum: *LoT: Dynamic Declustering of TSB-Tree Nodes for Parallel Access to Temporal Data*. In Proceedings of the Fifth International Conference on Extending Database Technology, 1996
- [177] T. Myrach: *Die Schlüsselproblematik bei der Umsetzung temporalen Konzepte in das relationale Datenmodell*. 1. Zobis-Workshop, Düsseldorf, Dez. 1995
- [178] S. S. Nair, S. K. Gadia: *Algebraic optimization in a relational model for temporal databases*. Technical Report TR–92–03, Computer Science Department, Iowa State University, Mai 1992
- [179] S. B. Navathe, R. Ahmed: *A temporal relational model and a query language*. UF–CIS Technical Report TR–85–16, Computer and Information Department, University of Florida, April 1986
- [180] S. B. Navathe, R. Ahmed: *TSQL—A language interface for historical databases*. In Proceedings of the Conference on Temporal Aspects in Information Systems, S.113–128, AFCET, Frankreich, Mai 1987
- [181] S. B. Navathe, R. Ahmed: *A temporal relational model and a query language*. Information Sciences, 49(2):147–175, 1989
- [182] Mehmet A. Orgun, Hausi A. Müller: *A Temporal Algebra Based on an Abstract*. In Proceedings of the ADC'93, 1993
- [183] R. Overmeyer, M. Stonebraker: *Implementation of a time expert in a database system*. In Proceedings of ACM SIGMOD International Conference on Management of Data, 12(3):51–59, April 1982
- [184] N. Pissinou, R. T. Snodgrass, R. Elmasri, I. S. Mumick, M. T. Özsu, B. Pernici, A. Segev, B. Theodoulidis, U. Dayal: *Towards an Infrastructure for Temporal Databases* Report of an Invitational ARPA/NSF Workshop, University of Arizona, Technical Report 94–01, Januar 1994
- [185] K. Ramamritham: *Time for Real-Time Temporal Databases*. In Proceedings of the International Workshop on an Infrastructure for Temporal Databases, Juni 1993
- [186] E. Rose, A. Segev: *TOODM—A temporal object-oriented data model with temporal constraints*. In Proceedings of the 10th International Conference on the Entity–Relationship Approach, S.205–229, Okt. 1991
- [187] E. Rose, A. Segev: *TO–Algebra—A temporal object-oriented algebra*. Technical Report LBL–32013, University of California, Berkeley, Jan. 1992
- [188] D. Rotem, A. Segev: *Physical organization of temporal databases*. In Proceedings of the International Conference on Data Engineering, S.547–553, Los Angeles, CA, Feb. 1987
- [189] R. Sadeghi: *A database query language for operations on historical data*. Ph. D. thesis, Dundee College of Technology, Dundee, Scotland, Dez. 1987
- [190] B. Salzberg, V. Tsotras: *A Comparison of Access Methods for Time Evolving Data*. Technical Report CATT–TR–94–81, Polytechnic University, Brooklyn, NY, 1994
- [191] N. L. Sarda: *Modelling of time and history data in database systems*. In Proceedings of the CIPS Congress 87 Winnipeg, S.15–20, CIPS, Mai 1987
- [192] N. L. Sarda: *An algorithm for time-rollback on historical relations*. Technical report, Division of Math, Engineering, and Computer Science, University of New Brunswick, Saint John, N. B. Canada, Mai 1988

- [193] N. L. Sarda: *Algebra and query language for a historical data model*. The Computer Journal, 33(1):11–18, Feb. 1990
- [194] N. L. Sarda: *Algebra and query language for a historical data model*. The Computer Journal, 33(1):11–18, Feb. 1990
- [195] N. L. Sarda: *Design of a historical database management system*. In Proceedings of the CSI Indore Chapter Conference, Aug. 1990
- [196] N. L. Sarda: *Extensions to SQL for historical databases*. IEEE Transactions on Knowledge and Data Engineering, 2(2):220–230, Juli 1990
- [197] N. L. Sarda: *Time-Grid: A file structure for historical databases*. Technical report, Indian Institute of Technology, April 1992
- [198] N. L. Sarda: *On handling future time*. Technical report, Indian Institute of Technology, Jan. 1992
- [199] A. Segev, A. Shoshani: *Logical modeling of temporal data*. In Proceedings of the ACM SIGMOD Annual Conference on Management of Data, S.454–466, San Francisco, CA, Mai 1987
- [200] A. Segev, A. Shoshani: *Modeling temporal semantics*. In Proceedings of the Conference on Temporal Aspects in Information Systems, S.47–60, Frankreich, Mai 1987
- [201] A. Segev, A. Shoshani: *Functionality of temporal data models and physical design implementations*. IEEE Database Engineering, 11(4):38–45, Dez. 1988
- [202] A. Segev, A. Shoshani: *The Representation of a Temporal Data Model in the Relational Environment*. Vol. 339, S.39–61, Springer Verlag, 1988
- [203] A. Segev, A. Shoshani: *The Representation of a Temporal Data Model in the Relational Environment*. In Proceedings of the Fourth International Conference on Statistical and Scientific Database Management, 1988
- [204] A. Segev, H. Gunadhi: *Event-join optimization in temporal relational databases*. In Proceedings of the Conference on Very Large Databases, S.205–215, Amsterdam, Holland, Aug. 1989
- [205] A. Segev, H. Gunadhi: *Processing event-joins in temporal databases*. Technical report, Lawrence Berkeley Laboratory, University of California, Berkeley, CA, 1992
- [206] A. Segev, H. Gunadhi, R. Chandra, J. G. Shanthikumar: *Selectivity estimation of temporal data manipulations*. Information Systems, 1992
- [207] A. Segev, C. S. Jensen, R. T. Snodgrass: *Report on The 1995 International Workshop on Temporal Databases*. SIGMOD Record, Vol. 24(4), Dez. 1995
- [208] J. Shifan: *An assessment of the temporal differentiation of attributes in the implementation of a temporally oriented DBMS*. Ph. D. thesis, Information Systems Area, Graduate School of Business Administration, New York University, Aug. 1986
- [209] A. Shoshani, K. Kawagoe: *Temporal data management*. In Proceedings of the Conference on Very Large Databases, S.79–88, Kyoto, Japan, Aug. 1986
- [210] R. Snodgrass: *The temporal query language TQel*. In Proceedings of the ACM Symposium on Principles of Database Systems, S.204–212, Waterloo, Ontario, Kanada, April 1984
- [211] R. Snodgrass, I. Ahn: *A taxonomy of time in databases*. In Proceedings of ACM SIGMOD International Conference of Data, S.236–246, Austin, TX, Mai 1985
- [212] R. Snodgrass (Ed.): *Research concerning time in databases: Project summaries*. ACM SIGMOD Record, 15(4):19–39, Dez. 1986
- [213] R. Snodgrass, I. Ahn: *Temporal databases*. IEEE Computer, 19(9):35–42, Sep. 1986
- [214] R. Snodgrass: *The temporal query language TQel*. ACM Transactions on Database Systems, 12(2):247–298, Juli 1987
- [215] R. Snodgrass, I. Ahn: *Partitioned storage for temporal databases*. Information Systems, 13(4):369–391, Dez. 1988
- [216] R. Snodgrass: *Special issue on temporal databases*. Data Engineering, 11(4), Dez. 1988
- [217] R. Snodgrass, I. Ahn: *Performance analysis of temporal queries*. Information Sciences, 49:103–146, Dez. 1989
- [218] R. Snodgrass: *Temporal databases: Status and research directions*. ACM SIGMOD Record, 19(4):83–89, Dez. 1990
- [219] R. Snodgrass: *Temporal databases*. In Proceedings of the International Conference on GIS: From Space to Territory, Vol. 629, Sep. 1992
- [220] R. T. Snodgrass: *Temporal Databases*. Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, ed. U. Formentini, S.22–64, Springer Verlag, 1992
- [221] R. Snodgrass: *Overview of the Special Section on Temporal Database Infrastructure*. University of Arizona, 1993
- [222] R. Snodgrass, S. Gomez, E. McKenzie: *Aggregates in the temporal query language TQel*. IEEE Transactions on Knowledge and Data Engineering, 1994
- [223] R. Snodgrass, The TSQL2 Language Design Committee: *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, Boston, Dordrecht, London, 1995
- [224] R. Snodgrass: *Temporal Object-Oriented Databases: A Critical Comparison*. Modern Database Systems, S. 386 – 408, ACM Press, 1995
- [225] X. Song, J. W. S. Liu: *How well Can Data Temporal Consistency be Maintained*. In Proceedings of the IEEE Symposium on Computer-Aided Control Systems Design, 1992
- [226] M. Soo: *Bibliography on temporal databases*. In Proceedings of ACM SIGMOD International Conference on Management of Data, 20(1):14–23, März 1991
- [227] M. Soo, R. Snodgrass: *Mixed calendar query language support for temporal constants*. TempIS Technical Report 29, Computer Science Department, University of Arizona, Tucson, Arizona, Mai 1992



- [228] M. Soo, R. Snodgrass: *Multiple calendar support for conventional database management systems*. Technical Report 92-7, Computer Science Department, University of Arizona, Feb. 1992
- [229] M. Soo, R. Snodgrass, C. Dyreson, C. S. Jensen, N. Kline: *Architectural extensions to support multiple calendars*. TempIS Technical Report 32, Computer Science Department, University of Arizona, Mai 1992
- [230] R. Stam, R. Snodgrass: *A bibliography on temporal databases*. Database Engineering, 7(4):231-239, Dez. 1988
- [231] R. Studer: *Modeling time aspects of information systems*. In Proceedings of the International Conference on Data Engineering, S.364-373, Los Angeles, CA, Feb. 1986
- [232] A. Uz Tansel, M. E. Arkun: *Aggregation operations in historical relational databases*. In Proceedings of the Third International Workshop on Statistical and Scientific Databases, Juli 1986
- [233] A. Uz Tansel, M. E. Arkun: *HQUEL, a query language for historical relational databases*. In Proceedings of the Third International Workshop on Statistical and Scientific Databases, Juli 1986
- [234] A. Uz Tansel: *Adding time dimension to relational model and extending relational algebra*. Information Systems, 11(4):343-355, 1986
- [235] A. Uz Tansel: *A statistical interface for historical relational databases*. In Proceedings of the International Conference on Data Engineering, S.538-546, Los Angeles, CA, Feb. 1987
- [236] A. Uz Tansel, M. E. Arkun, G. Özsoyoğlu: *Time-by-example query language for historical databases*. IEEE Transactions on Software Engineering, 15(4):464-478, April 1989
- [237] A. Uz Tansel, L. Garnett: *Nested historical relations*. In Proceedings of ACM SIGMOD International Conference on Management of Data, S.284-293, Mai 1989
- [238] A. Uz Tansel: *A historical query language*. Information Sciences, 53:101-133, 1991
- [239] A. Uz Tansel: *On Roth, Korth, and Silberschatz's extended algebra and calculus languages for nested relational databases*. ACM Transactions on Database Systems, 17(2), Juni 1992
- [240] A. Uz Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, R. T. Snodgrass: *Temporal databases: theory, design, and implementation*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, 1993
- [241] V. J. Tsotras, A. Kumar: *Temporal Database Bibliography Update*. Department of Electrical Engineering & Computer Science, Polytechnic University, Brooklyn, New York, 1996
- [242] A. Tuzhilin, J. Clifford: *A temporal relational algebra as a basis for temporal relational completeness*. In Proceedings of the Conference on Very Large Databases, S.13-23, Brisbane, Australien, Aug. 1990
- [243] Hartmut Wedekind: *Are the Terms „Version“ and „Variant“ Orthogonal to One Another*. Universität Erlangen-Nürnberg
- [244] G. Wiederhold, S. Jajodia, W. Litwin: *Dealing with granularity of time in temporal databases*. In Proceedings of the Third Nordic Conference on Advanced Information Systems Engineering, Trondheim, Norway, Mai 1991

## Internetadressen zu temporalen Datenmodellen

<http://wwwdb.informatik.uni-rostock.de/~bg/te1.html> Verschiedene Technische Reports, Studien- und Diplomarbeiten zum Thema Temporale Datenmodelle (deutsch) incl. Links

<http://sunsite.informatik.rwth-aachen.de/dblp/db/conf/tdb/index.html> Workshop on Temporal Databases-Page, einige Links

<http://www.daimi.aau.dk/~kock/TempDB95/> Workshop on Temporal Databases-Homepage, viele Informationen, u. a. Adresse für das Bestellen der Proceedings

<http://www.mathematik.uni-marburg.de/~seeger/zeit.html> Anfragebearbeitung in historischen Datenbanken , Uni Marburg

<http://www.informatik.uni-trier.de/~ley/db/index.html> Bibliographie Server Database and Logic Programming mit vielen Links (Tagungen, Proceedings, Veröffentlichungen, Autoren)

## Internetadressen zu Geoinformationssystemen

<http://www.mathematik.uni-marburg.de/~seeger/geo.html> Anfragebearbeitung in Geo-Datenbanksystemen, Uni Marburg

<http://www.inf.fu-berlin.de/~voisard/gisprod.html> GIS-Produkte, Links zu ESRI, Intergraph u. a., Geometrische Metadaten, Open GIS

<http://www.regis.berkeley.edu:80/index.html> REGIS-Homepage (University of Berkeley), viele nützliche Informationen und Links

<http://www.bio-geo.uni-karlsruhe.de/gis/gis.htm> Großartige GIS-Seite, gut sortierte Liste der meisten GIS-Server weltweit, Veranstaltungen, Forschungsberichte, Adressen, Jobs

<http://www.hdm.com/gis3.htm> GIS-Seite mit umfangreichem Material und Links (hauptsächlich Amerika)

# Erklärung

Ich versichere, diese Diplomarbeit selbständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt zu haben.

---

Jan Ramsch, Leipzig, den 4. April 1997

# A Datenstrukturtabellen

## A.1 IfAG–interne Objektarten

Objektart	ATT	Attributwert	Kommentar
9211* Bebaute Ortslage	BEB	1000 offene Bebauung 9997 geschlossene Bebauung	OAR nicht im OK
2121 Bergbau -Sign.	QUE*	2000* wird als Punktobjekt erfaßt	Punktobjekt im OK nicht vorgesehen
2301 Tagebau -Sign.	QUE*	2000* wird als Punktobjekt erfaßt	Punktobjekt im OK nicht vorgesehen
9213* Friedhofsignatur			Signatur zusätzlich
2221 Stadionsignatur	QUE*	2000* wird als Punktobjekt erfaßt	Punktobjekt im OK nicht vorgesehen
9230* Torfstichsignatur			Signatur zusätzlich
2315 Gebäudesignatur	QUE*	1000* wird immer als Punktobjekt erfaßt	auch wenn Fläche gefordert
3101 Straße	KLS*	1000* Autobahnsignatur 1001* Autobahnsignatur ohne Verstärkung 3000* IA-Signatur 3500* IA/IB-Signatur ohne Verstärkung 4000* IB-Signatur 5000* II-Signatur 5500* Anliegerstraßen	Attribut KLS im OK nicht enthalten
3106 Fahrbahn	KLS*	2000* halbe Autobahnsign. 3000* IA-Signatur 5000* II-Signatur 6000* Abfahrt eine Richtung	Attribut KLS im OK nicht enthalten
3201 Schienenbahn	GLZ DGA*	9000* mehrgleisig 1000* Normalspur 2000* Schmalspur	nicht im OK für Signatur Gleisanschlusses
3302 Flugplatz	QUE*	1000* wird immer als Punktobjekt erfaßt	auch wenn Fläche gefordert
3501 Bahnhofsanlage	QUE* ADB*	1000* wird immer als Punktobjekt erfaßt 1000* an Vollspurbahn 2000* an Schmalspurbahn 3000* an U-Bahn	auch wenn Fläche gefordert für Signatur notwendig
3514 Brücke, Unterführung, Überführung	AUS*	1000* Brücke 2000* Überführung 3000* Unterführung 9997* trifft nicht zu	für Signatur notwendig
9351* Brückenschenkel Straße			im OK nicht vorgesehen

Objektart	ATT	Attributwert	Kommentar
9353* Brückenschenkel Bahn			im OK nicht vorgesehen
3523* Schifffahrtszeichen	KON*	5005* Schifffahrtszeichen	OAR existiert nicht im OK200
9411* Heidesignatur			OAR nicht im OK
9412* Moorsign. Gras			OAR nicht im OK
9413* Moorsign. Strich			OAR nicht im OK
9414* Sumpfsign. Strich			OAR nicht im OK
5101 Strom, Fluß, Bach	BRG SFK	1* Quellbach 9000* schiffbar	Kl. 1 nicht im OK Wert nicht im OK
6102 DGM-Höhenlinie	HLA HLT*	9999* sonstige 1000* Zähllinie 2000* Hauptlinie 3000* Hilfslinie lang 4000* Hilfslinie kurz	Wert nicht im OK Attribut für Signatur notwendig
6104 Besonderer Geländepunkt	DGP* AGP	1000* normale Darstellung 2000* abgeschwächte Darst. 8000* Wasserspiegelpunkt	Attribut für Signatur Wert nicht im OK
6204 Böschung, Kliff	KNT* BSA*	3000* ausgepr. Oberkante 4000* unausgeprägte Oberk. 9998* beliebige Oberkante 1000* freie Böschung 2000* Grubenböschung 4000* Tagebauböschung	Attribut für Signatur  Attribut für Signatur
7299 Grenze	GDS*	1000* L=dick, B=dick 2000* L=dünn, B=dick 3000* L=dünn, B=dünn 4000* L=dick 5000* L=duenn 6000* B=dick 7000* B=dünn	Attribut für Signatur

\* – Neu eingeführtes Element

Namen der neu eingeführten Attribute:

ADB	Art der Bahn	QUE	Qualität der Erfassung
KLS	Straßensignatur	DGA	Darstellungsart des Gleisanschlusses
HLT	Darstellung der Höhenlinie	DGP	Darstellung des Geländepunktes
KNT	Ausprägung der Böschungskante	GDS	Grenzdarstellung
BSA	Böschungsart für künstliche Böschungen		
AUS	Ausprägung der Brücke, Überführung, Unterführung		

## A.2 Attributklassen

OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.
2111	DBD	3	2332	HHO	3	6205	OFM	1	5101	PRB	3
5102	PRB	3	5111	PRB	3	5112	PRB	3	3512	LGO	3
2101	FLB	3	2332	HHO	3	3532	DMS	3	1002	GN	1
1002	KN	1	1002	ZN	1	1003	GN	1	1003	KN	1
1003	ZN	1	2101	OFT	3	2101	VOH	3	2101	HNN	3
2101	BWD	3	2101	DBD	3	2111	HBB	3	2111	BWD	3
2111	HNN	3	2111	VOH	3	2112	HBI	3	2112	BWD	3
2112	DBD	3	2112	HNN	3	2112	VOH	3	2113	HBB	3
2113	BWD	3	2113	DBD	3	2113	HNN	3	2113	VOH	3
2114	HBB	3	2114	BWD	3	2114	HNN	3	2114	DBD	3
2114	VOH	3	2121	TFO	3	2121	BWD	3	2121	DBD	3
2122	ABF	3	2122	HHO	3	2122	OFM	3	2122	BWD	3
2122	DBD	3	2122	HNN	3	2122	VOH	3	2123	BWD	3
2123	DBD	3	2123	HNN	3	2123	VOH	3	2124	GN	1
2124	ZN	1	2124	KN	1	2124	ZUS	3	2124	OFT	3
2124	BWD	3	2124	DBD	3	2124	HNN	3	2124	VOH	3
2125	BWD	3	2125	DBD	3	2125	HNN	3	2125	VOH	3
2126	BWD	3	2126	DBD	3	2126	HNN	3	2126	VOH	3
2127	ZUS	3	2127	BWD	3	2127	DBD	3	2127	HNN	3
2127	VOH	3	2128	ZUS	3	2128	BWD	3	2128	DBD	3
2128	HNN	3	2128	VOH	3	2129	GN	1	2129	ZN	1
2129	KN	1	2129	NTZ	3	2129	ZUS	3	2129	OFT	3
2129	BWD	3	2129	DBD	3	2129	HNN	3	2129	VOH	3
2130	GN	1	2130	ZN	1	2130	KN	1	2130	FKT	3
2130	OFT	3	2130	PRO	3	2130	ZUS	3	2130	BWD	3
2130	DBD	3	2130	HNN	3	2130	VOH	3	2131	GN	1
2131	ZN	1	2131	KN	1	2131	OFT	3	2131	BWD	3
2131	DBD	3	2131	HNN	3	2131	VOH	3	2132	GN	1
2132	ZN	1	2132	KN	1	2132	OFT	3	2132	BWD	3
2132	DBD	3	2132	HNN	3	2132	VOH	3	2133	ZUS	3
2133	BWD	3	2133	DBD	3	2133	HNN	3	2133	VOH	3
2134	GN	1	2134	ZN	1	2134	KN	1	2134	ZUS	3
2134	OFT	3	2134	BWD	3	2134	DBD	3	2134	HNN	3
2134	VOH	3	2135	HHO	3	2135	OFL	3	2135	OFM	3
2135	ZUS	3	2135	BWD	3	2135	DBD	3	2135	HNN	3
2135	VOH	3	2201	NTZ	3	2201	BWD	3	2201	DBD	3
2201	HNN	3	2201	VOH	3	2202	BWD	3	2202	DBD	3
2202	HNN	3	2202	VOH	3	2211	GN	1	2211	ZN	1
1001	GN	1	1001	KN	1	1001	ZN	1	2101	GN	1
2101	KN	1	2101	ZN	1	2111	BEB	3	2111	GN	1
2111	KN	1	2111	OFT	3	2111	ZN	1	2112	BEB	3

Tabelle 16: Objektattribute und ihre Zuordnung zu Attributklassen, Teil 1

OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.
2112	FKT	3	2112	GN	1	2112	KN	1	2112	OFT	3
2112	ZN	1	2113	BEB	3	2113	GN	1	2113	KN	1
2113	OFT	3	2113	STA	3	2113	ZN	1	2114	BEB	3
2114	FKT	3	2114	GN	1	2114	KN	1	2114	OFT	3
2114	ZN	1	2121	GN	1	2121	KN	1	2121	OFL	3
2121	OFT	3	2121	PRO	3	2121	ZN	1	2121	ZUS	3
2122	GN	1	2122	KN	1	2122	OFL	3	2122	OFT	3
2122	ZN	1	2122	ZUS	3	2123	GN	1	2123	KN	1
2123	OFT	3	2123	ZN	1	2123	ZUS	3	2125	GN	1
2125	KN	1	2125	NTZ	3	2125	OFT	3	2125	PRO	3
2125	ZN	1	2125	ZUS	3	2126	GN	1	2126	KN	1
2126	OFT	3	2126	PEG	3	2126	ZN	1	2126	ZUS	3
2127	GN	1	2127	KN	1	2127	OFT	3	2127	ZN	1
2128	GN	1	2128	KN	1	2128	OFT	3	2128	PRO	3
2128	ZN	1	2133	GN	1	2133	KN	1	2133	OFT	3
2133	PEG	3	2133	ZN	1	2135	ABF	3	2135	GN	1
2135	KN	1	2135	ZN	1	2201	GN	1	2201	KN	1
2201	OFT	3	2201	ZN	1	2202	GN	1	2202	KN	1
2202	OFT	3	2202	ZN	1	2213	GN	1	2213	KN	1
2213	OFT	3	2213	ZN	1	2221	GN	1	2221	KN	1
2221	OFT	3	2221	SPO	3	2221	ZN	1	2228	GN	1
2228	KN	1	2228	ZN	1	2229	GN	1	2229	KN	1
2229	OFT	3	2229	ZN	1	2301	GN	1	2301	KN	1
2301	OFT	3	2301	PRO	3	2301	TFO	3	2301	ZN	1
2301	ZUS	3	2302	GN	1	2302	HHO	3	2302	KN	1
2302	PRO	3	2302	ZN	1	2313	DMS	3	2313	GN	1
2313	HHO	3	2313	KN	1	2313	KON	3	2313	OFL	3
2313	PRO	3	2313	ZN	1	2314	GN	1	2314	KN	1
2314	OFT	3	2314	ZN	1	2315	BRO	3	2315	GFK	3
2315	GN	1	2315	HHO	3	2315	HNN	3	2315	KN	1
2315	WDM	3	2315	ZN	1	2315	ZUS	3	2316	DMS	3
2316	FKT	3	2316	GN	1	2316	HHO	3	2316	HNN	3
2316	KN	1	2316	WDM	3	2316	ZN	1	2317	HHO	3
2317	HNN	3	2319	DMS	3	2319	GN	1	2319	KN	1
2319	ZN	1	2320	GN	1	2320	KN	1	2320	ZN	1
2331	ATP	3	2331	GN	1	2331	KN	1	2331	ZN	1
2332	GN	1	2332	KN	1	2332	ZN	1	2344	GN	1
2344	KN	1	2344	SPO	3	2344	ZN	1	2346	GN	1
2346	KN	1	2346	ZN	1	3101	BDI	3	3101	BDU	3
3101	FKT	3	3101	GN	1	3101	IBD	2	3101	KN	1
3101	WDM	2	3101	ZN	1	3101	ZUS	3	3102	BEF	3
3102	FKT	2	3102	GN	1	3102	KN	1	3102	ZN	1
3102	ZUS	3	3103	FKT	3	3103	GN	1	3103	KN	1

Tabelle 17: Objektattribute und ihre Zuordnung zu Attributklassen, Teil 2

OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.
3103	ZN	1	3104	GN	1	3104	KN	1	3104	ZN	1
3106	BDI	3	3106	BDU	3	3106	BRF	3	3106	FKT	3
3106	GN	1	3106	IBD	2	3106	KN	1	3106	WDM	2
3106	ZN	1	3106	ZUS	3	3201	BKT	3	3201	ELK	3
3201	GLZ	3	3201	GN	1	3201	KN	1	3201	KON	3
3201	NTZ	3	3201	SPW	3	3201	ZN	1	3201	ZUS	3
3202	BKT	3	3202	GN	1	3202	KN	1	3202	ZN	1
3301	FKT	3	3301	GN	1	3301	KN	1	3301	NTZ	3
3301	OFT	3	3301	ZN	1	3302	FKT	3	3302	GN	1
3302	KN	1	3302	NTZ	3	3302	OFT	3	3302	ZN	1
3302	ZUS	3	3303	BRO	3	3303	FKT	2	3303	GN	1
3303	KN	1	3303	OFM	3	3303	ZN	1	3304	GN	1
3304	KN	1	3304	ZN	1	3401	GN	1	3401	KN	1
3401	NTZ	3	3401	SFK	3	3401	ZN	1	3402	GN	1
3402	KN	1	3402	NTZ	3	3402	SFK	3	3402	TID	3
3402	ZN	1	3403	FKT	3	3403	GN	1	3403	KN	1
3403	ZN	1	3501	BFK	3	3501	FKT	3	3501	GN	1
3501	KN	1	3501	ZN	1	3501	ZUS	3	3502	GN	1
3502	KN	1	3502	ZN	1	3503	GN	1	3503	KN	1
3503	KON	3	3503	ZN	1	3511	GN	1	3511	KN	1
3511	ZN	1	3513	BRO	3	3513	GN	1	3513	KN	1
3513	ZN	1	3514	GN	1	3514	HHO	3	3514	KN	1
3514	ZN	1	3531	GN	1	3531	KN	1	3531	ZN	1
3532	GN	1	3532	KN	1	3532	PRO	1	3532	ZN	1
3541	FKT	3	3541	GN	1	3541	HHO	3	3541	KN	1
3541	ZN	1	3542	GN	1	3542	HHO	3	3542	KN	1
3542	ZN	1	3543	FKT	3	3543	GN	1	3543	HHO	3
3543	KN	1	3543	KON	3	3543	ZN	1	4104	GN	1
4104	KN	1	4104	ZN	1	4105	GN	1	4105	KN	1
4105	ZN	1	4106	GN	1	4106	HYD	3	4106	KN	1
4106	ZN	1	4107	GN	1	4107	KN	1	4107	NTZ	3
4107	ZN	1	4109	GN	1	4109	KLT	3	4109	KN	1
4109	ZN	1	4111	GN	1	4111	KN	1	4111	ZN	1
4120	GN	1	4120	KN	1	4120	OFM	3	4120	ZN	1
5101	BRG	3	5101	GN	1	5101	HYD	3	5101	KN	1
5101	OFL	3	5101	PRB	3	5101	SFK	3	5101	TID	3
5101	WDM	3	5101	ZN	1	5102	BRG	3	5102	GN	1
5102	KN	1	5102	PRB	3	5102	SFK	3	5102	TID	3
5102	WDM	3	5102	ZN	1	5102	ZUS	3	5103	BRG	3
5103	GN	1	5103	KN	1	5103	OFL	3	5103	TID	3
5103	ZN	1	5104	GN	1	5104	KN	1	5104	ZN	1
5105	GN	1	5105	KN	1	5105	ZN	1	5111	GN	1

Tabelle 18: Objektattribute und ihre Zuordnung zu Attributklassen, Teil 3



OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.
5111	KN	1	5111	PRB	3	5111	ZN	1	5112	ENT	3
5112	GN	1	5112	HYD	3	5112	KN	1	5112	PRB	3
5112	ZN	1	5121	GN	1	5121	KN	1	5121	ZN	1
5201	GN	1	5201	KN	1	5201	ZN	1	5203	GN	1
5203	KN	1	5203	ZN	1	5302	BRO	3	5302	GN	1
5302	KN	1	5302	KON	3	5302	OFM	3	5302	ZN	1
5302	ZUS	3	5303	BRO	3	5303	FKT	3	5303	GN	1
5303	KN	1	5303	KON	3	5303	LGO	3	5303	ZN	1
5303	ZUS	3	5321	BRO	3	5321	KON	3	5321	OFM	3
6101	GN	1	6101	KN	1	6101	ZN	1	6102	GN	1
6102	HHL	1	6102	HLA	3	6102	KN	1	6102	ZN	1
6103	AGL	1	6103	GN	1	6103	KN	1	6103	ZN	1
6104	AGP	3	6104	GN	1	6104	KN	1	6104	PKH	3
6104	ZN	1	6199	GN	1	6199	KN	1	6199	ZN	1
6201	FKT	3	6201	GN	1	6201	KN	1	6201	ZN	1
6202	GN	1	6202	KN	1	6202	ZN	1	6203	FKT	3
6203	GN	1	6203	KN	1	6203	ZN	1	6204	BOK	3
6204	ENT	3	6204	GN	1	6204	KN	1	6204	OFM	3
6204	ZN	1	6205	BOK	1	6205	ENT	1	6205	GN	1
6205	KN	1	6205	ZN	1	6206	GN	1	6206	KN	1
6206	ZN	1	6207	GN	1	6207	KN	1	6207	OFM	3
6207	ZN	1	6211	GN	1	6211	KN	1	6211	OFL	3
6211	WDM	3	6211	ZN	1	6212	GN	1	6212	KN	1
6212	ZN	1	6213	GN	1	6213	KN	1	6213	ZN	1
6214	GN	1	6214	KN	1	6214	ZN	1	6215	GN	1
6215	KN	1	6215	ZN	1	7101	ADM	2	7101	EWZ	3
7101	FLA	3	7101	GN	1	7101	KN	1	7101	STG	3
7101	ZFK	3	7101	ZN	1	7102	FOK	3	7102	GN	1
7102	KN	1	7102	ZN	1	7103	EWZ	1	7103	FLA	1
7103	GN	1	7103	KN	1	7103	ZN	1	7201	FLB	3
7201	GN	1	7201	KN	1	7201	LTP	3	7201	ZN	1
7202	FLB	3	7202	GN	1	7202	KN	1	7202	LTP	3
7202	ZN	1	7211	FLA	3	7211	GN	1	7211	KN	1
7211	ZN	1	7299	APG	2	7299	GN	1	7299	KN	1
7299	LZB	3	7299	ZN	1	7299	ZUS	3	7301	FLA	3
7301	GN	1	7301	KN	1	7301	SZE	3	7301	ZN	1
7302	FLA	3	7302	GN	1	7302	KN	1	7302	ZN	1
7305	FLA	3	7305	GN	1	7305	KN	1	7305	ZN	1
7311	FLA	3	7311	GN	1	7311	KN	1	7311	ZN	1
7312	FLB	3	7312	GN	1	7312	KN	1	7312	SZE	3
7312	ZN	1	7402	GN	1	7402	KN	1	7402	ZN	1
7403	FLA	3	7403	GN	1	7403	KN	1	7403	ZN	1

Tabelle 19: Objektattribute und ihre Zuordnung zu Attributklassen, Teil 4

OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.
7405	FLA	3	7405	GN	1	7405	KN	1	7405	NTZ	3
7405	ZN	1	2211	KN	1	2211	OFT	3	2211	BWD	3
2211	DBD	3	2211	HNN	3	2211	VOH	3	2212	GN	1
2212	ZN	1	2212	KN	1	2212	OFT	3	2212	BWD	3
2212	DBD	3	2212	HNN	3	2212	VOH	3	2213	REL	3
2213	ZUS	3	2213	BWD	3	2213	VOH	3	2221	NTZ	3
2221	BWD	3	2221	DBD	3	2221	HNN	3	2221	VOH	3
2222	GN	1	2222	ZN	1	2222	KN	1	2222	NTZ	3
2222	SPO	3	2222	OFT	3	2222	BWD	3	2222	DBD	3
2222	HNN	3	2222	VOH	3	2223	GN	1	2223	ZN	1
2223	KN	1	2223	SSZ	3	2224	GN	1	2224	ZN	1
2224	KN	1	2224	NTZ	3	2225	GN	1	2225	ZN	1
2225	KN	1	2225	NTZ	3	2225	OFT	3	2226	GN	1
2226	ZN	1	2226	KN	1	2226	NTZ	3	2226	OFT	3
2227	GN	1	2227	ZN	1	2227	KN	1	2227	FKT	3
2227	NTZ	3	2227	OFT	3	2228	OFM	3	2228	NTZ	3
2230	GN	1	2230	ZN	1	2230	KN	1	2230	OFT	3
2301	BWD	3	2301	DBD	3	2302	OFM	3	2303	GN	1
2303	ZN	1	2303	KN	1	2303	FKT	3	2303	NTZ	3
2303	OFM	3	2310	GN	1	2310	ZN	1	2310	KN	1
2310	BRO	3	2310	DMS	3	2310	HHO	3	2310	LGO	3
2311	GN	1	2311	ZN	1	2311	KN	1	2311	HHO	3
2311	ZUS	3	2311	OFT	3	2312	GN	1	2312	ZN	1
2312	KN	1	2312	HHO	3	2312	OFT	3	2312	BWD	3
2312	DBD	3	2312	HNN	3	2312	VOH	3	2313	BRO	3
2313	LGO	3	2313	OFM	3	2314	HHO	3	2314	VEG	3
2314	PRO	3	2315	GSZ	3	2315	LGO	3	2315	NTZ	3
2315	OFL	3	2315	OFM	3	2315	BDA	3	2315	BST	3
2315	VOH	3	2316	BRO	3	2316	LGO	3	2316	NTZ	3
2316	OFM	3	2316	ZUS	3	2317	GN	1	2317	ZN	1
2317	KN	1	2317	BRO	3	2317	DMS	3	2317	LGO	3
2317	OFM	3	2318	GN	1	2318	ZN	1	2318	KN	1
2318	BRO	3	2318	DHU	3	2319	BRO	3	2319	FKT	3
2319	HYD	3	2319	LGO	3	2319	NTZ	3	2319	TFO	3
2319	ZUS	3	2319	KON	3	2319	OFT	3	2319	WAS	3
2320	BRO	3	2320	DMS	3	2320	KON	3	2320	LGO	3
2320	ZUS	3	2321	GN	1	2321	ZN	1	2321	KN	1
2321	DMS	3	2321	HHO	3	2321	OFT	3	2321	HNN	3
2322	GN	1	2322	ZN	1	2322	KN	1	2322	HHO	3
2322	ZUS	3	2322	OFT	3	2323	GN	1	2323	ZN	1
2323	KN	1	2323	HHO	3	2323	ZUS	3	2323	OFT	3
2324	GN	1	2324	ZN	1	2324	KN	1	2324	BRO	3

Tabelle 20: Objektattribute und ihre Zuordnung zu Attributklassen, Teil 5

OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.
2324	DMS	3	2324	HHO	3	2324	KON	3	2324	LGO	3
2324	ZUS	3	2324	TRK	3	2325	GN	1	2325	ZN	1
2325	KN	1	2325	BRO	3	2325	DMS	3	2325	HHO	3
2325	LGO	3	2325	PRO	3	2325	ZUS	3	2325	OFT	3
2326	GN	1	2326	ZN	1	2326	KN	1	2326	BRO	3
2326	DMW	3	2326	LGO	3	2326	FKT	3	2326	OFT	3
2327	GN	1	2327	ZN	1	2327	KN	1	2327	BRO	3
2327	DMS	3	2327	FKT	3	2327	HHO	3	2327	LGO	3
2327	ZUS	3	2327	OFT	3	2328	GN	1	2328	ZN	1
2328	KN	1	2328	NTZ	3	2328	OFT	3	2331	BRO	3
2331	DMS	3	2331	HHO	3	2331	LGO	3	2331	WDM	3
2331	EPO	3	2331	OFT	3	2331	BWD	3	2331	DBD	3
2331	HNN	3	2331	VOH	3	2332	BRO	3	2332	DMS	3
2332	LGO	3	2332	WDM	3	2332	OFT	3	2333	GN	1
2333	ZN	1	2333	KN	1	2333	BRO	3	2333	DMS	3
2333	HHO	3	2333	LGO	3	2333	OFM	3	2333	WDM	3
2334	GN	1	2334	ZN	1	2334	KN	1	2334	DMS	3
2334	HHO	3	2334	OFM	3	2341	GN	1	2341	ZN	1
2341	KN	1	2341	HHO	3	2341	KON	3	2341	OFT	3
2342	GN	1	2342	ZN	1	2342	KN	1	2342	SPO	3
2342	OFT	3	2343	GN	1	2343	ZN	1	2343	KN	1
2343	HHO	3	2343	KON	3	2343	OFT	3	2344	BRO	3
2344	OFT	3	2345	GN	1	2345	ZN	1	2345	KN	1
2345	NTZ	3	2346	BRO	3	2346	HHO	3	2346	KON	3
2346	OFT	3	2346	HNN	3	2351	GN	1	2351	ZN	1
2351	KN	1	2351	BRO	3	2351	FKT	3	2351	VOH	3
2351	OFT	3	2352	GN	1	2352	ZN	1	2352	KN	1
2352	VOH	3	2352	OFT	3	3101	BFS	3	3101	FSZ	3
3101	BRV	3	3101	NTZ	2	3101	OFM	3	3102	BRF	3
3102	BRV	3	3102	KON	3	3102	OFM	3	3102	WDM	3
3103	NTZ	3	3103	OFM	3	3104	FTR	1	3104	AFT	1
3105	GN	1	3105	ZN	1	3105	KN	1	3105	BDI	3
3105	BDU	3	3105	BRV	3	3105	IBD	2	3105	NTZ	2
3105	WDM	2	3106	BFS	3	3106	FSZ	3	3106	OFM	3
3201	BDU	3	3201	BKP	3	3201	BRF	3	3201	GLS	3
3201	BKL	3	3201	BNA	3	3201	OFT	3	3202	DHU	3
3202	NTZ	3	3202	VOH	3	3203	GN	1	3203	ZN	1
3203	KN	1	3204	GN	1	3204	ZN	1	3204	KN	1
3204	BKP	3	3204	BRV	3	3204	GLZ	3	3204	OFT	3
3205	GN	1	3205	ZN	1	3205	KN	1	3205	BDU	3
3205	BKT	2	3205	ELK	3	3205	GLS	3	3205	KON	3
3205	NTZ	3	3205	SPW	3	3205	ZUS	3	3205	BNA	3

Tabelle 21: Objektattribute und ihre Zuordnung zu Attributklassen, Teil 6

OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.
3205	BKL	3	3205	OFT	3	3301	STZ	3	3301	VOH	3
3301	HNN	3	3301	BWD	3	3301	DBD	3	3302	BWD	3
3302	DBD	3	3302	HNN	3	3302	VOH	3	3303	ZUS	3
3304	OFM	3	3304	FKT	3	3401	FKT	3	3401	HFZ	3
3401	BWD	3	3401	DBD	3	3401	HNN	3	3401	VOH	3
3501	BDU	3	3501	BSG	3	3501	OFT	3	3501	BWD	3
3501	DBD	3	3501	HNN	3	3501	VOH	3	3502	BWD	3
3502	DBD	3	3502	HNN	3	3502	VOH	3	3511	BWD	3
3511	DBD	3	3511	HNN	3	3511	VOH	3	3512	GN	1
3512	ZN	1	3512	KN	1	3512	BRO	3	3512	FKT	3
3512	LGO	3	3512	NTZ	3	3512	KON	3	3512	TRF	3
3513	DHO	3	3513	ROZ	3	3514	BRO	3	3514	DHO	3
3514	DHU	3	3514	KON	3	3514	OFM	3	3514	TRF	3
3514	ZUS	3	3515	GN	1	3515	ZN	1	3515	KN	1
3515	BEF	3	3515	BRO	3	3515	LGO	3	3516	GN	1
3516	ZN	1	3516	KN	1	3516	DHO	3	3516	SIC	3
3521	GN	1	3521	ZN	1	3521	KN	1	3521	BRO	3
3521	DMS	3	3521	HHO	3	3521	KON	3	3521	LGO	3
3521	OFM	3	3522	GN	1	3522	ZN	1	3522	KN	1
3522	FKT	2	3522	KNA	3	3522	KNE	3	3522	KON	3
3522	STM	3	3522	SW1	3	3523	GN	1	3523	ZN	1
3523	KN	1	3523	BRO	3	3523	HHO	3	3523	KON	3
3523	LGO	3	3523	ZUS	3	3523	OFT	3	3524	GN	1
3524	ZN	1	3524	KN	1	3524	FKT	3	3531	FKT	3
3531	OFL	3	3531	SPG	3	3532	DHU	3	3532	NTZ	3
3532	OFL	3	3532	ZUS	3	3532	KON	3	3532	OFT	3
3533	GN	1	3533	ZN	1	3533	KN	1	3533	BRO	3
3533	DHU	3	3533	VOH	3	3533	OFL	3	3533	OFT	3
3541	BRO	3	3541	DMS	3	3541	LGO	3	3541	OFM	3
3541	KON	3	3542	DMR	3	3542	OFT	3	3543	BRO	3
3543	DMS	3	3543	LGO	3	3543	NTZ	3	3543	OFT	3
4101	GN	1	4101	ZN	1	4101	KN	1	4101	VEG	3
4101	BDQ	3	4101	OFT	3	4102	GN	1	4102	ZN	1
4102	KN	1	4102	FKT	3	4102	VEG	3	4102	BDQ	3
4102	OFT	3	4103	GN	1	4103	ZN	1	4103	KN	1
4103	FKT	3	4103	VOH	3	4103	BDQ	3	4103	OFT	3
4104	OFT	3	4105	VEG	3	4105	OFT	3	4106	VEG	3
4106	OFT	3	4107	VEG	3	4107	VOH	3	4107	BMA	3
4107	OFT	3	4107	ZUS	3	4108	GN	1	4108	ZN	1
4108	KN	1	4108	VEG	3	4108	VOH	3	4108	OFT	3
4109	VOH	3	4109	BDQ	3	4109	OFT	3	4110	GN	1
4110	ZN	1	4110	KN	1	4110	ENT	3	4110	OFT	3

Tabelle 22: Objektattribute und ihre Zuordnung zu Attributklassen, Teil 7

OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.	OAR	ATT	Kl.
4199	GN	1	4199	ZN	1	4199	KN	1	4201	GN	1
4201	ZN	1	4201	KN	1	4201	HHO	3	4201	VEG	3
4201	WDM	3	4201	BMA	3	4202	GN	1	4202	ZN	1
4202	KN	1	4202	VEG	3	4202	VOH	3	4202	BMA	3
4202	MBA	3	4203	GN	1	4203	ZN	1	4203	KN	1
4203	BRO	3	4203	KON	3	4203	VOH	3	5101	VEG	3
5101	OFT	3	5102	VEG	3	5102	OFT	3	5103	FKT	3
5103	HYD	3	5103	VEG	3	5103	WDM	3	5105	HYD	3
5105	KON	3	5105	VEG	3	5105	FKT	3	5105	QUA	3
5105	WAS	3	5111	TID	3	5112	HST	3	5112	VEG	3
5112	WDM	3	5112	FKT	3	5112	OFT	3	5202	GN	1
5202	ZN	1	5202	KN	1	5202	HHO	3	5203	BRO	3
5203	HHO	3	5301	GN	1	5301	ZN	1	5301	KN	1
5301	DMS	3	5301	KON	3	5302	HHO	3	5302	FKT	3
5303	HHO	3	5303	KMZ	3	5303	OFT	3	5303	HNN	3
5304	GN	1	5304	ZN	1	5304	KN	1	5304	BRO	3
5304	HHO	3	5304	LGO	3	5304	ZUS	3	5311	GN	1
5311	ZN	1	5311	KN	1	5311	PEH	3	5311	KON	3
5321	GN	1	5321	ZN	1	5321	KN	1	5321	VOH	3
6101	GIW	3	6201	BDK	3	6201	BRD	3	6201	WHO	3
6201	WDM	3	6203	BDK	3	6203	WDM	3	6204	BHO	3
6204	ZUS	3	6204	WDM	3	6205	BHO	1	6205	WDM	1
6207	HHO	3	6211	BRO	3	6211	DMS	3	6211	HHO	3
6211	HNN	3	6211	LGO	3	7203	GN	1	7203	ZN	1
7203	KN	1	7203	FLB	3	7302	ZUS	3	7303	GN	1
7303	ZN	1	7303	KN	1	7303	FLA	3	7303	SZO	3
7303	ZUS	3	7304	GN	1	7304	ZN	1	7304	KN	1
7304	FLA	3	7304	SZO	3	7304	ZUS	3	7305	SZO	3
7311	FKT	1	7311	NRS	3	7311	SZE	3	7311	SZQ	3
7311	ZUS	3	7311	FLB	3	7401	GN	1	7401	ZN	1
7401	KN	1	7401	FLA	3	7402	FLA	3	7404	GN	1
7404	ZN	1	7404	KN	1	7404	ENT	3	7404	FLA	3
7404	SZO	3	9211	BEB	3	2315	QUE	3	3101	KLS	3
3106	KLS	3	3501	ADB	3	6102	HLT	3	6104	DGP	3
6204	KNT	3	6204	BSA	3	7299	GDS	3	2121	QUE	3
2301	QUE	3	2221	QUE	3	3514	AUS	3	3302	QUE	3
3501	QUE	3	3201	DGA	3						

Tabelle 23: Objektattribute und ihre Zuordnung zu Attributklassen, Teil 8

### A.3 SQL–Aufrufe zur Generierung der DST

```
create table objektart
```

```
(oar          integer    primary key,  
 name        char(60)  not null unique,  
 typ25       char(1)   not null,  
 typ200      char(1),  
 typ1000     char(1),  
 netz        char(1),  
 dlm         char(1)   not null,  
 ueberlagern char(1),  
 quelle      char(1)   not null,  
 check (typ25 in ('l','p','f','g','k','a','b','c','d')),  
 check (typ200 in ('l','p','f','g','k','a','b','c','d')),  
 check (typ1000 in ('l','p','f','g','k','a','b','c','d')),  
 check (netz  in ('e','s','g')),  
 check (dlm   in ('a','b','c')),  
 check (ueberlagern in ('j')),  
 check (quelle in ('a','i'))  
);
```

```
create table ist_Teil_von
```

```
(oar1          integer,  
 oar2          integer,  
 foreign key (oar1) references objektart(oar),  
 foreign key (oar2) references objektart(oar)  
);
```

```
create table liegt_ueber
```

```
(oar1          integer,  
 oar2          integer,  
 foreign key (oar1) references objektart(oar),  
 foreign key (oar2) references objektart(oar)  
);
```

```
create table attribut
```

```
(attribut      char(3)   primary key,  
 name         char(90)  not null,  
 typ          char(1)   not null,  
 stellenzahl  integer,  
 me           char(3),  
 me_c1000     char(3),  
 art          char(1)   not null,  
 belegung     char(1)   not null,  
 check (typ   in ('i','c','b','f','d','n')),  
 check (me    in ('m','km','dm','ha','qkm','l/s','kV','%')),  
 check (me_c  in ('m','km','dm','ha','qkm','l/s','kV')),  
 check (art   in ('s','k','a')),
```

```

    check (belegung      in ('j','n'))
);

create table hat_attribut
(oar                integer,
 attribut          char(3),
 klasse           char(1) not null,
 kategorie        char(1) not null,
 dlm              char(1) not null,
 quelle           char(1) not null,
 check (klasse     in ('1','2','3')),
 check (kategorie  in ('1','2','3')),
 check (dlm        in ('a','b','c','d','e')),
 check (quelle     in ('a','i')),
 foreign key (oar) references objektart,
 foreign key (attribut) references attribut,
 unique(oar,attribut,klasse)
);

create table wert
(attribut          char(3),
 wert             integer,
 bedeutung        char(70),
 dlm              char(1),
 check (dlm       in ('a','b','c','d','e')),
 foreign key (attribut) references attribut,
 primary key (attribut,wert,dlm)
);

create table hat_wert
(oar                integer,
 attribut          char(3),
 wert             integer,
 dlm              char(1),
 quelle           char(1),
 check (quelle     in ('a','i')),
 foreign key (oar) references objektart,
 foreign key (attribut,wert,dlm) references wert,
 unique(oar,attribut,wert,dlm)
);

```

## A.4 Komponenten der Datenstrukturtabellen

Tabelle	Spalte	Datentyp	Werte	Kommentar
objektart	oar	integer		Objektartschlüssel laut OK
	name	char(60)		Objektartname laut OK
	typ25 typ200 typ1000	char(1)	p l f g k a b c d	punktförmig linienförmig flächenförmig Gitter Komplexobjekt linien- oder punktförmig flächen- oder punktförmig flächen- oder linienförmig flächen-, linien- oder punktförmig
	netz	char(1)	e s g	Eisenbahnnetz Straßennetz Gewässernetz
	dln	char(1)	a b c	DLM 25 DLM 25 und DLM 200 DLM 25, DLM 200 und DLM 1000
	ueberlagern	char(1)	j NULL	Objekt darf andere überlagern darf nicht überlagern
	quelle	char(1)	a i	ATKIS-Objektart IfAG-interne Objektart
	hat_attribut	oar	integer	
attribut		char(3)		Fremdschlüssel zu attribut.attribut
klasse		char(1)	1 2 3	DLM-übergreifendes Attribut Objektabhängiges Attribut Objektteilattribut
kategorie		char(1)	1, 2, 3	Kategorie laut OK
dln		char(1)	a b c d e f g	DLM 25 DLM 25 und DLM 200 DLM 25, DLM 200 und DLM 1000 DLM 200 DLM 1000 DLM 200 und DLM 1000 DLM 25 und DLM 1000
quelle		char(1)	a i	ATKIS-Attribut IfAG-internes Attribut

Tabelle 24: Komponenten der Datenstrukturtabellen, Teil 1



Tabelle	Spalte	Datentyp	Werte	Kommentar
attribut	attribut	char(3)		Attributkennung laut OK
	name	char(90)		volle Bezeichnung laut OK
	typ	char(1)	c i b f d	Charakter & String Integer Binär Gleitkommazahl Dezimal
	stellenzahl	integer		Anzahl der Stellen bei Charakter
	me	char(3)	m,km,dm	Maßeinheit des Attributes laut OK
	me_1000	char(3)	l/s,qkm ha,kV,%	Alternative Maßeinheit für DLM 1000
	art	char(1)	s k a	Attribute mit tatsächlichem Wert Tatsächliche, aber klassifizierte Werte Kodierte Attribute (4- stellige Codes)
	belegung	char(1)	j n	Attribut wird in jetziger Bearbeitungsstufe erfaßt Attribut wird noch nicht erfaßt
hat_wert	oar	integer		Fremdschlüssel zu objektart.oar
	attribut	char(3)		Fremdschlüssel zu wert.attribut
	wert	integer		Fremdschlüssel zu wert.wert
	dml	char(1)		Fremdschlüssel zu wert.dml
	quelle	char(1)	a i	ATKIS-Attributwert IfAG-interner Attributwert
wert	attribut	char(3)		Fremdschlüssel zu attribut.attribut
	wert	integer		tatsächlicher, klassifizierter Wert oder kodierter Wert
	bedeutung	char(70)		Klassenbezeichnung oder Bedeutung eines kodierten Wertes
	dml	char(1)	a b c d e f g	DLM 25 DLM 25 und DLM 200 DLM 25, DLM 200 und DLM 1000 DLM 200 DLM 1000 DLM 200 und DLM 1000 DLM 25 und DLM 1000
liegt_ueber	oar1	integer		Fremdschlüssel zu objektart.oar
	oar2	integer		Fremdschlüssel zu objektart.oar
ist_teil_von	oar1	integer		Fremdschlüssel zu objektart.oar
	oar2	integer		Fremdschlüssel zu objektart.oar

Tabelle 25: Komponenten der Datenstrukturtabellen, Teil 2

## A.5 Unstimmigkeitenkatalog

Dieser Abschnitt beinhaltet alle bei der Analyse der OK für den Aufbau der DST zu Tage getretenen Fehler und Unstimmigkeiten. Jeder Eintrag hat folgenden prinzipiellen Aufbau:

Objektartkode (4–stelliger Schlüssel)      Objektart  
 Attributkode (3–buchstabiger Schlüssel)    Attribut  
 Attributwertkode (4–stellig)                Attributwert  
 weitere Attributwerte

Einzelne Komponenten können entfallen, wenn sie für den Fehler unwichtig sind.

In der vierten Spalte ist die Art des Fehlers mit Hilfe der in Abschnitt 6.3 eingeführten Fehlernummern angegeben.

Diese Aufstellung bezieht sich auf den Stand der OK vom 1.11.1995.

OK 25		OK 200		OK 1000		Fehler
3501	Bahnhofsanlage	3501	Bahnhofsanlage	3501	Bahnhofsanlage	2
FKT	Funktion	FKT	Funktion	FKT	Funktion	
2201	Personenverkehr	2201	Personenverkehr, Auto im Reisezug	2201	Personenverkehr	
2207	Auto im Reisezug					
2203	Güterverkehr	2203	Güterverkehr, Containerverkehr, Huckepackverkehr	2203	Güterverkehr	
2205	Containerverkehr					2,3
2206	Huckepackverkehr					
2126	Kraftwerk	2126	Kraftwerk			
PEG	Primärenergie	PEG	Primärenergie			
1300	Sonne	1300	Sonne			
1800	Verbrennung	1800	Verbrennung, Kohle, Öl, Gas, Müll, Abfall			
1801	Kohle					
1802	Öl					
1803	Gas					
1804	Müll, Abfall					
9999	sonstige	9999	sonstige			
2133	Heizwerk	2133	Heizwerk			
PEG	Primärenergie	PEG	Primärenergie			
1300	Sonne					
1800	Verbrennung					
1801	Kohle					
1802	Öl					
1803	Gas					
1804	Müll, Abfall	1804	Müll, Abfall			
9999	sonstige	9999	Sonne, Verbrennung, Kohle, Öl, Gas, sonstige			

OK 25		OK 200		OK 1000	Fehler
5302	Talsperre, Wehr	5302	Talsperre, Wehr		2
6204	Böschung, Kliff	6204	Böschung, Kliff		
OFM	Oberflächenmaterial	OFM	Oberflächenmaterial		
1000	Erde	1000	Erde		
5321	Uferbefestigung	5321	Uferbefestigung		
OFM	Oberflächenmaterial	OFM	Oberflächenmaterial		
1000	Erde	1000	Erde, Pflanzenbewuchs		
1100	Pflanzenbewuchs				
5302	Talsperre, Wehr	5302	Talsperre, Wehr		2
OFM	Oberflächenmaterial	OFM	Oberflächenmaterial		
2000	Gestein	2000	Gestein, Fels		
4120	Vegetationslose Fläche	4120	Vegetationslose Fläche		
6204	Böschung, Kliff	6204	Böschung, Kliff		
6207	Stützmauer	6207	Stützmauer		
OFM	Oberflächenmaterial	OFM	Oberflächenmaterial		
2100	Fels	2100	Fels		
4120	Vegetationslose Fläche	4120	Vegetationslose Fläche		2,4
5321	Uferbefestigung	5321	Uferbefestigung		
OFM	Oberflächenmaterial	OFM	Oberflächenmaterial		
2200	Steine, Schotter	2200	Steine, Schotter, Geröll		
2300	Geröll				
6204	Böschung, Kliff	6204	Böschung, Kliff		
6207	Stützmauer	6207	Stützmauer		
OFM	Oberflächenmaterial	OFM	Oberflächenmaterial		
2200	Steine, Schotter	2200	Steine, Schotter, Geröll		
2300	nicht vorhanden				
6205	Böschung, Kliff (komplex)	6205	Böschung, Kliff (komplex)		
OFM	nicht vorhanden	OFM	Oberflächenmaterial		
		2200	Steine, Schotter		
2121	Bergbaubetrieb	2121	Bergbaubetrieb		1,2
PRO	Produkt (Abbaugut)	PRO	Produkt (Abbaugut)		
1000	Erde	1000	Erde, Abraum, Geröll		
1005	Abraum				
2300	Geröll				
2301	Tagebau, Steinbruch, Grube	2301	Tagebau, Steinbruch, Grube		
PRO	Produkt (Abbaugut)	PRO	Produkt (Abbaugut)		
1000	Erde	1000	Erde, Abraum, Geröll		
1005	nicht vorhanden				
2300	nicht vorhanden				
2302	Halde, Aufschüttung	2302	Halde, Aufschüttung		
PRO	Produkt (Aufschüttungs- material)	PRO	Produkt (Aufschüttungs- material)		
1000	Erde	1000	Erde, Abraum, Geröll		
1005	Abraum				
2300	Geröll				

OK 25		OK 200		OK 1000		Fehler
2125 PRO 4000	Lager, Depot Produkt (Lagergut) Treib-, Brennstoffe	2125 PRO 4000	Lager, Depot Produkt (Lagergut) Treib-, Brennstoffe, Erdöl, Gas			1,2
4100 4200	Erdöl Gas					
2313 PRO 4000 4100 4200	Vorratsbehälter, Speicherbauwerk Produkt (Speicherinhalt) Treib-, Brennstoffe Erdöl Gas	2313 PRO 4000 4100 4200	Vorratsbehälter, Speicherbauwerk Produkt (Speicherinhalt) Treib-, Brennstoffe Erdöl Gas			
2315 ZUS 1200	Gebäude Zustand außer Betrieb, stillgelegt	2315 ZUS 1200	Gebäude Zustand außer Betrieb, stillge- legt, verlassen (,Wüstung‘)			2
1800	verlassen (,Wüstung‘)					
3101 ZUS 1200	Straße Zustand außer Betrieb, stillgelegt	3101 ZUS 1200	Straße Zustand außer Betrieb, stillgelegt			
5xxx BRG 42 200 9999	Gewässer Breite des Gewässers nicht vorhanden nicht vorhanden nicht vorhanden	5xxx BRG 42 200 9999	Gewässer Breite des Gewässers über 30 m bis 42 m nicht vorhanden über 42 m	5xxx BRG 42 200 9999	Gewässer Breite des Gewässers bis 42 m über 42 m bis 200 m über 200 m	3
7101 ADM 4007	Verwaltungseinheit Administrative Funktion ursprünglich gemein- defreies Gebiet	7101 ADM 4007	Verwaltungseinheit Administrative Funktion ursprünglich gemein- defreies Gebiet	7101 ADM 4007	Verwaltungseinheit Administrative Funktion selbstständige Gemeinde	3
7311 SZE	Wasserschutzgebiet, Heilquellenschutzge- biet Schutzzone allgemein oder gegen qualitati- ve Beeinträchtigung	7311 SZE	Wasserschutzgebiet, Heilquellenschutzge- biet Schutzzone			1
5101 5112 HYD 2000	Strom, Fluß, Bach Binnensee, Stausee, Teich Hydrologisches Merkmal nicht ständig was- serführend	5101 5112 HYD 2000	Strom, Fluß, Bach Binnensee, Stausee, Teich Hydrologisches Merkmal nicht ständig was- serführend	5101 5112 HYD 2000	Strom, Fluß, Bach Binnensee, Stausee, Teich Hydrologisches Merkmal nicht ständig peri- odisch wasserführend	3
3103 FKT 9999	Platz Funktion sonstige	3103 FKT 9999	Platz Funktion Festplatz, sonstige			3
3101 WDM 9999	Straße Widmung sonstige	3101 WDM 9999	Straße Widmung Gemeindestraße, sonstige			3

OK 25		OK 200		OK 1000		Fehler
3106 WDM	Fahrbahn nicht vorhanden	3106 WDM 9999	Fahrbahn Widmung Gemeindestraße, sonstige			3,4
3531	Freileitung	3531	Freileitung	3531	Kabelleitung	1
7301 7312 SZO	Nationalpark Lärmschutzbereich Schutzzone	7301 7312 SZE	Nationalpark Lärmschutzbereich Schutzzone			1
6100	Digitales Geländehöhenmodell (DGM)	6100	Gelände	6100	Digitales Geländehöhenmodell (DGM)	1
6200	Besondere Geländeoberflächenformen	6200	Besondere Geländeformen	6200	Besondere Geländeoberflächenformen	1
2315 GFK 1303	Gebäude Gebäudefunktion nicht vorhanden	2315 GFK 1303	Gebäude Gebäudefunktion Hotel/Motel	2315 GFK 1303	Gebäude Gebäudefunktion nicht vorhanden	4
3102 3501 FKT 9999	Weg Bahnhofsanlage Funktion sonstige	3102 3501 FKT 9999	Weg Bahnhofsanlage Funktion nicht vorhanden	3501 FKT 9999	Bahnhofsanlage Funktion sonstige	4
6211 OFL 1700	Felsen, Felsblock, Felsnadel Lage zur Erdoberfläche nicht vorhanden	6211 OFL 1700	Felsen, Felsblock, Felsnadel Lage zur Erdoberfläche nicht vorhanden	6211 OFL 1700	Felsen, Felsblock, Felsnadel Lage zur Erdoberfläche unter der Wasseroberfläche (bez. auf MW)	4
3501 FKT 9999	Bahnhofsanlage Funktion sonstige	3501 FKT 9999	Bahnhofsanlage Funktion nicht vorhanden	3501 FKT 9999	Bahnhofsanlage Funktion sonstige	4
3106 WDM BDU BDI IBD	Fahrbahn nicht vorhanden nicht vorhanden nicht vorhanden	3106 WDM BDU BDI IBD	Fahrbahn Widmung Verkehrsbedeutung überörtlich Verkehrsbedeutung innerörtlich Internationale Bedeutung	3106 WDM BDU BDI IBD	Fahrbahn Widmung Verkehrsbedeutung überörtlich nicht vorhanden Internationale Bedeutung	4
3302 ZUS	Flugplatz, Landeplatz nicht vorhanden	3302 ZUS	Flugplatz, Landeplatz Zustand	3302 ZUS	Flugplatz, Landeplatz Zustand	4
5101 5102 5111 5112 PRB	Strom, Fluß, Bach Kanal (Schiffahrt) Meer Binnensee, Stausee, Teich nicht vorhanden	5101 5102 5111 5112 PRB	Strom, Fluß, Bach Schiffahrt Meer Binnensee, Stausee, Teich Periodische Beschränkung	5101 5102 5111 5112 PRB	Strom, Fluß, Bach Schiffahrt Meer Binnensee, Stausee, Teich Periodische Beschränkung	4
3532 DMS	Rohrleitung, Pipeline Durchmesser des Objekts	3532 DMS	Rohrleitung, Pipeline nicht vorhanden	3532 DMS	Rohrleitung, Pipeline Durchmesser des Objekts	4
BRF	allgemein: Breite der Fahrbahn stetiges Attribut	BRF	Breite der Fahrbahn klassifiziertes Attribut	BRF	Breite der Fahrbahn klassifiziertes Attribut	5

OK 25, 200 und 1000		Fehler
2121	Bergbaubetrieb	1
2301	Tagebau, Grube, Steinbruch	
PRO	Produkt (Abbaugut)	
2125	Lager, Depot	
PRO	Produkt (Lagergut)	
2128	Förderanlage	
PRO	Produkt (Fördergut)	
2130	Fabrikanlage, Werksanlage	
PRO	Produkt (Herstellungsgut)	
2302	Halde, Aufschüttung	
PRO	Produkt (Aufschüttungsmaterial)	
2313	Vorratsbehälter, Speicherbauwerk	
PRO	Produkt (Speicherinhalt)	
2314	Absetzbecken, Schlammteich, Erdfaulbecken, Rieselfeld	
PRO	Produkt	
2315	Gebäude	
2316	Turm	
PRO	Produkt (Baumaterial)	
2325	Pumpe, Pumpstelle	
PRO	Produkt (Transportgut)	
3532	Rohrleitung, Pipeline	
PRO	Produkt, Transportgut	

## B Tools für die ATKIS–Datenbank

Dieser Anhang enthält in kompakter Form eine Aufstellung und Erläuterung der wichtigsten Tools, die im Zusammenhang mit dieser Arbeit am IfAG in Leipzig entstanden sind. Alle Programme sind in der ARC/INFO–Makrosprache AML geschrieben. Die Kommunikation mit den unter Informix verwirklichten Tabellen der DST und der ATKIS–Datenbank erfolgte über den Modul Database Integrator des ARC/INFO–Paketes mittels SQL–Anweisungen.

An sinnvollen Stellen wurden den Toolbeschreibungen stark vereinfachte Programmablaufpläne und Screenshots der erzeugten graphischen Oberflächen beigelegt. Sie repräsentieren den Stand zum Abschluß dieser Arbeit.

### B.1 db\_report

Programmaufruf:

```
DB_REPORT <INIT> <report_filename> <25|200|1000> {debug}
DB_REPORT <HELP>
```

Die Routine db\_report gibt in übersichtlicher, OK–ähnlicher Form den wesentlichen Inhalt der DST bezüglich eines DLM wieder.

Als Parameter werden ein Dateiname <report\_filename> und die Nummer eines DLM (25, 200 oder 1000) verlangt. Die Ergebnisse der Recherche werden in die spezifizierte Datei geschrieben, welche im aktuellen Arbeitsverzeichnis angelegt wird.

Existiert bereits eine Datei mit dem selben Namen, so wird diese ohne Rückfrage überschrieben.

Als Eingaben für <DLM> sind 25, 200 und 1000 erlaubt. In einem Durchlauf ist nur die Ausgabe bezüglich eines DLM möglich.

Der optionale Parameter {debug} ist nur für Entwickler und evt. als Kontrolle bei fehlerhaftem Programmablauf gedacht. Er bewirkt ausführliche Bildschirmausgaben aller Aktivitäten des Programmes, insbesondere aller abgesetzten SQL–Statements und wichtiger interner Variablen.

Folgende Hinweise und bekannte Mängel sind zu beachten:

- Das Programm kann nur DST der Version 1.8 auswerten.
- In seltenen Fällen kann ein SQL–Ausdruck vom Informix nicht ausgewertet werden (Error: sql-statement failed to parse ...) Das starten des Programms unter ARCEDIT schafft hier im allgemeinen Abhilfe. Ursache ist die Beschränkung des char–Datentyps auf 255 Zeichen unter ARC. Sollte db\_report auch dann noch mit dieser

Fehlermeldung abbrechen, so liegt wahrscheinlich ein struktureller Fehler in den DST vor, eventuell wurde eine inkompatible Version der DST verwendet.

- Bricht das Programm mit einer der Fehlermeldungen „Unknown database: dst“, „Connect failed: ...“ o. ä. ab, so ist entweder:
  - der Datenbankserver unerreichbar (z.B. Netzfehler),
  - der Datenbankserver inaktiv oder
  - die Konfigurationsdatei `dst.dbs` nicht vorhanden oder falsch.

Die Konfigurationsdatei `dst.dbs` enthält wichtige Parameter zur Kommunikation mit einem externen Datenbanksystem und wird vom Database Integrator ausgewertet. In einer typischen ARC/INFO-Umgebung liegt diese Datei im Verzeichnis

```
/esri/arcexe70/database/
```

und hat bei der Verwendung von Informix den folgenden Inhalt:

```
ARC/INFO 7.0 - Informix Server
$ARCHOME/programs/dbi_informix
dst
```

## B.2 create\_atkis\_ex

Programmaufruf:

```
CREATE_ATKIS_EX <INIT> <DLM> {sql-file} {log-file} {debug}
CREATE_ATKIS_EX <HELP>
```

Diese Routine erzeugt die DLM-Datenstruktur unter Informix und die benötigten Cover unter ARC/INFO unter Zuhilfenahme von Informationen aus den DST. Das Programm läuft vollständig autark.

Als Parameter werden <DLM> (25, 200 oder 1000) und optional jeweils ein Dateiname für das SQL- und das Logfile verlangt.

Das SQL-file enthält alle während des Programmlaufes abgesetzten `create table`-Anweisungen. Sollte während der Ausführung eines SQL-Kommandos ein Fehler aufgetreten sein, der zum Abbruch des Programmes führte, so ist auch das fehlererzeugende Statement im SQL-File enthalten.

Das Logfile enthält allgemeine Informationen über die abgearbeiteten Objektarten und eine kleine Statistik.



Wird einer oder beide Dateinamen nicht spezifiziert, so wird für das SQL-File der Dateiname `create_atkis_ex.sql` und für das Logfile `create_atkis_ex.log` verwendet.

Existiert bereits eine Datei mit dem selben Namen, so wird diese ohne Rückfrage überschrieben.

Der optionale Parameter `{debug}` ist nur für Entwickler und evt. als Kontrolle bei fehlerhaftem Programmablauf gedacht. Er bewirkt ausführliche Bildschirmausgaben aller Aktivitäten des Programmes, insbesondere aller abgesetzten SQL-Statements und wichtiger interner Variablen.

Folgende Hinweise und bekannte Mängel sind zu beachten:

- Das Programm kann nur DST der Version 1.8 auswerten
- Die Datenbank ATKIS muß verfügbar und leer sein. Der Benutzer, der das Programm `create_atkis_ex` startet, muß Schreib- und Änderungsrechte für die Datenbank ATKIS besitzen (Nutzer muß DBA sein).
- Die Cover `%d1m%hcover`, `%d1m%gcover` und `%d1m%pcover`, wobei `%d1m%` für a, b oder c steht (entsprechend der DLM 25, 200 und 1000), müssen im aktuellen Workspace existieren.
- Das Programm kann nur unter ARC gestartet werden.
- Bricht das Programm mit der Fehlermeldung „`unable to restore quoted string ...`“ ab, so wurde höchstwahrscheinlich ein zu langer SQL-Ausdruck (≥850 Zeichen) erzeugt. Abhilfe kann hier nur ein Eingriff in den Quelltext schaffen. Entweder man ändert das Programm dahingehend, das für jede zu erzeugende Tabelle mehrere, kürzere SQL-Anweisungen erzeugt werden (mit relativ hohem Aufwand verbunden), oder man geht wie folgt vor:
  1. Auskommentieren aller DBMSEXECUTE-Anweisungen im Programm (8 Stück).
  2. Starten des Programmes (sollte der Fehler dennoch auftreten, so bleibt nur die erstgenannte Möglichkeit).
  3. Manuelles Ausführen des erzeugten SQL-Files unter Informix (oder dem benutzten RDBMS).
- Bricht das Programm mit einer der Fehlermeldungen „`Unknown database: dst`“, „`Connect failed: ...`“ o. ä. ab, so ist entweder:
  - der Datenbankserver unerreichbar (z.B. Netzfehler),
  - der Datenbankserver inaktiv oder

- die Konfigurationsdatei `dst.dbs`<sup>1</sup> nicht vorhanden oder falsch.
- Analoges gilt für „Unknown database: atkis“.
- Kann ein SQL-Ausdruck vom Informix nicht ausgewertet werden (`Error: sql-statement failed to parse ...`), so liegt wahrscheinlich ein struktureller oder inhaltlicher Fehler in den DST vor, eventuell wurde eine falsche Version der DST verwendet.

### B.3 atkisoj

Programmaufruf:

```
ATKISOJ <INIT> <DLM> {debug}  
ATKISOJ <HELP>
```

---

<sup>1</sup> Details zu `dst.dbs` siehe Anhang B.1.