

UNIGIS MSc



Universität Salzburg
Geographical Information Science and Systems

Master Thesis

Sven-Henrik Kleber
(UNIGIS 2001, U832)

Datenmigration – alles Standard ?

Betreuer: Prof. Dr. Josef Strobl

Januar 2003

Erklärung

Hiermit versichere ich, dass ich die vorliegende Master Thesis selbstständig verfasst und keine anderen als die angegebenen Quellen verwendet habe.

Sven-Henrik Kleber

Netphen, im Januar 2003

Zusammenfassung

Die vorliegende Arbeit beschreibt ein Prozedere, welches ein größerer Industriekonzern in einem Zeitraum von etwa eineinhalb Jahren durchlaufen hat, um einen historisch gewachsenen Datenbestand von einem veralteten Datenbanksystem in ein zeitgemäßes Oracle-Umfeld zu überführen.

Zu Beginn wird das Unternehmen mit seiner Entstehung und der heutigen Struktur vorgestellt. Dabei wird die bereits vorhandene Software und die damit verbundene Datenarchitektur beschrieben. Anschließend wird auf die Motivation und die damit verbundene Zielsetzung des Unternehmens für eine Datenmigration eingegangen. Es sprachen einige Gründe, wie z.B. die Schaffung von Auskunftsarbeitsplätzen und der Wunsch nach verschiedenen Analysemöglichkeiten dafür, den Datenbestand in ein aktuelles DBMS zu migrieren.

Im weiteren Verlauf wird die GIS-Architektur des Projektes erläutert. In heutigen GI-Systemen hat sich eine dreischichtige Architektur bestehend aus Datenhaltung, Middleware und Client als vorteilhaft herausgestellt.

Es werden derzeit verschiedene Ansätze zur Datenmigration diskutiert. Generell gilt es zu entscheiden, ob eine Direktschnittstelle sinnvoll ist, oder ob die Konvertierung über ein Zwischenformat vorteilhafter ist. Der gewählte Ansatz, ein Zwischenformat einzusetzen, bietet einige Vorteile gegenüber einer Direktübertragung.

In den letzten Abschnitten geht es um die Frage, ob die angestrebten Ziele und Visionen des Unternehmens erreicht wurden. Dazu wird als originärer Nutzen beschrieben, welche Möglichkeiten es gibt, die Daten an verschiedenen Orten zur Verfügung zu stellen, was durch eine Nutzung der IMS-Technologie realisiert wird. Eine weitere Nutzung der umgesetzten Daten durch Drittsysteme wird beschrieben, indem mit dem exemplarisch gewählten GI-System GeoMedia Professional auf den migrierten Datenbestand zugegriffen wird. Es wird gezeigt, dass eine standardisierte Datenhaltung für eine Datennutzung durch beliebige Systeme nicht ausreichend ist, und dass sich die Systeme in der Realität in keiner Weise derart interoperabel darstellen, wie stets für sie geworben wird. Erst durch die ständige Anpassung der Metatabellen können mehrere GI-Systeme auf den selben Datenbestand lesend und schreibend zugreifen.

Abstract

This Master Thesis describes a workflow which a major industrial enterprise went through over a period of one and a half year in order to transform their old continuously grown data base from the existing outdated form into a modern Oracle-environment.

At the beginning history and present structure of the industrial enterprise are introduced. Existing software and architecture of the data base are described. Then motivation and intensions of the industrial enterprise for the migration of the data base are pointed out. There were many different reasons (e.g. creation of information terminals or desire of analysing data in several ways) to migrate the data base into a modern DBMS.

The following step shows the GIS-architecture of the project. In present GI-systems the 3-tier-architecture has been established consisting of three parts database, middleware and client.

At the moment there are different approaches in discussion for data migration. First you have to decide between a direct interface or a conversion using an interformat. In this project the approach of an interformat is realised because there are several advantages over the direct interface.

The last part discusses if intensions and visions of the industrial enterprise have been achieved. In order to answer this question it is described how the data base could be provided at different places. This is realised with the help of the IMS-technology. The point of original use of the data base is followed by the next step to use the data base by third party systems. Exemplarily the GI-system GeoMedia Professional is used to access the migrated data base. It is shown that a standard data management is inadequate to be used as a data base of any GI-system. Furthermore the systems are not interoperable how it is promoted by the software producers. Many systems can only use the same data base for reading and writing if the meta tables are updated constantly by an own application.

Inhaltsverzeichnis

1	EINLEITUNG	1
2	AUSGANGSSITUATION	3
2.1	Firma Dynamit Nobel	3
2.2	Software Topodat von C-Plan	4
2.3	Zielsetzung	8
2.4	Zusammenfassung	11
3	GIS-ARCHITEKTUR DES PROJEKTES	12
3.1	Datenhaltung der Spatial-Architektur	12
3.1.1	Modellierung	14
3.1.2	Abfragemodell	19
3.1.3	Räumliche Indizierung	19
3.2	Middleware	20
3.3	Client	29
3.4	Zusammenfassung	32
4	ANSÄTZE ZUR DATENMIGRATION	34
4.1	Direktschnittstelle	34
4.2	Zwischenformat	35
4.3	Bewertung	36
5	UMSETZUNG DES GEWÄHLTEN ANSATZES	39
5.1	Ablauf der Datenübernahme	39
5.2	Datenübernahme in einen Zwischenuser	41
5.3	Datenübernahme in den Enduser	47
5.4	Zusammenfassung	54

6	EINSATZ DER ERZEUGTEN DATEN	55
6.1	Originärer Nutzen der Daten	55
6.2	Externer Nutzen der Daten	59
6.2.2	Metadatenstruktur von GeoMedia Professional	63
6.2.3	Zugriff auf den Datenbestand	69
6.3	Zusammenfassung	83
7	ABSCHLUSSBEWERTUNG UND AUSBLICK	85
8	LITERATURVERZEICHNIS	88
	ANHANG A	89
	ANHANG B	91
	ANHANG C	98

Abbildungsverzeichnis

Abbildung 2-1: Karte aus Topodat.....	7
Abbildung 3-1: 3-tier-Architektur	12
Abbildung 3-2: Logische Datenmodell von Oracle	14
Abbildung 3-3: Räumliche Abfragemodell.....	19
Abbildung 3-4: Quadtree-Methode	20
Abbildung 3-5: Bestandteile der Topobase-Umgebung	21
Abbildung 3-6: Tabelle TB_Dictionary	23
Abbildung 3-7: Tabelle TB_Constraint.....	24
Abbildung 3-8: Tabelle TB_Relations	24
Abbildung 3-9: Beziehungen in der Topobase	25
Abbildung 3-10: Verknüpfung der Layer-Tabellen.....	26
Abbildung 3-11: Layer-Tabellen.....	26
Abbildung 3-12:Komplette Struktur einer Topobase	28
Abbildung 4-1: Austauschformat CITRA	36
Abbildung 4-2: Datentransfer in FME	37
Abbildung 5-1: Ablauf der Datenübernahme	39
Abbildung 5-2: Datei Topo2RDB.inp	41
Abbildung 5-3: Programm Topo2RDB.....	41
Abbildung 5-4: Zeitfenster in Topo2RDB	42
Abbildung 5-5: Tabelle Topo_punkt	44
Abbildung 5-6: Tabelle Topo_linkopf	44
Abbildung 5-7: Tabelle Topo_makro.....	45
Abbildung 5-8: Programm TB_EXCHANGE	46
Abbildung 5-9: Neue Topobase erstellen.....	49
Abbildung 5-10: Erstellung eines Topics.....	49
Abbildung 5-11: Erstellen einer Tabelle	50
Abbildung 5-12: Struktur eines Topics.....	51
Abbildung 5-13: Tabelle Exchange_konv_table_geom	52
Abbildung 5-14: Tabelle Exchange_konv_table_attr.....	53
Abbildung 6-1: Von der Datenquelle zum Client.....	56
Abbildung 6-2: Ansicht einer mwf-Datei	57
Abbildung 6-4: Relationale Modell von GeoMedia	61
Abbildung 6-5: Objektrelationale Modell von GeoMedia	62
Abbildung 6-7: Warehouse-Verbindungsassistent	70
Abbildung 6-8: Parameter für Datenbankverbindung	70
Abbildung 6-9: Objektklasseneintrag hinzufügen	71
Abbildung 6-10: Definition der Spatial Objects	73
Abbildung 6-11: Wertebereich der Objekte	74
Abbildung 6-12: Featuretabelle in Oracle.....	75
Abbildung 6-13: Datenbankverbindung	76
Abbildung 6-14: Datenbank-Dienstprogramm.....	76
Abbildung 6-15: Objektklassen-Metadaten einfügen	77
Abbildung 6-16: Objektklassenauswahl	77
Abbildung 6-17: Oracle Spatial Analyzer	79
Abbildung 6-18: Karte aus Oracledatenbank in GeoMedia Professional.....	80

Tabellenverzeichnis

Tabelle 2.1: Struktur der elemda.top.....	5
Tabelle 3.1: Linien_SDOLAYER.....	15
Tabelle 3.2: Linien_SDODIM	16
Tabelle 3.3: Linien_SDOGEOM.....	17
Tabelle 3.4: Linien_SDOINDEX.....	17
Tabelle 3.5: Tabellentypen.....	24
Tabelle 3.6: Featuretabelle "Linien"	27
Tabelle 3.7: Geometrietabelle "Linien_geo"	27
Tabelle 6.1: Vergleich Topobase und GeoMedia Pro	69

1 Einleitung

In der heutigen GIS-Welt sind Schlagworte wie Objektorientierung, Komponententechnologie und Internet bereits selbstverständlich geworden. Hinzu kommen speziellere Begriffe wie Interoperabilität oder verteilte Geodaten. Solche Begriffe entstammen aus der rasanten Entwicklung von neuen Technologien und der damit verbundenen Kurzlebigkeit von Systemen. Aus dieser Kurzlebigkeit heraus müssen heute unterschiedlichste strategische Einflüsse erkannt und bewertet werden. Bei den Einflüssen spielt die allgemeine Softwaretechnik und die GIS-technischen Anforderungen ebenso eine wichtige Rolle wie Planungs- und Investitionssicherheit zu gewährleisten. Solche Begriffe entstammen aber auch dem Ruf der GIS-Welt nach Standardisierungen, wie sie beispielsweise durch das OpenGis Konsortium vertreten und erarbeitet werden.

In diesem Zusammenhang ist es von erheblicher Bedeutung, über die zukünftigen Möglichkeiten nicht die Vergangenheit zu vernachlässigen. Früher getätigte Investitionen müssen dabei ebenso berücksichtigt werden, wie die Planungen und Festlegungen für die Zukunft, in der ein Investitionsschutz immer deutlicher zum zentralen Thema wird. Die vergangenen Investitionen beziehen sich dabei weniger auf ein konkretes Softwaresystem, sondern hauptsächlich auf die Erfassung und Speicherung von (Geo)-Daten, da diese in den meisten Fällen den weitaus höheren Kostenanteil darstellen. Daraus folgt, dass ein Datenbestand ebenso in der Lage sein muss sich weiterzuentwickeln, wie die Softwaretechnologie, die diese Daten umgibt. Bei einem Umstieg auf eine neue Technologie stellt sich also nicht nur die Frage nach einer geeigneten Systemwahl für die Zukunft, sondern auch die Frage nach der Migration des vorhandenen Datenbestandes in die neue Systemumgebung. Dieser Schritt, auch Datenmigration genannt, stellt oftmals die schwierigste Aufgabe bei einem Systemwechsel dar. Die Komplexität der Datenstrukturen, die Architektur der GIS-Software oder auch die Qualität der GIS-Daten nehmen Einfluss auf den Schwierigkeitsgrad einer Datenmigration.

Obwohl heute nahezu alle GIS-Softwarehersteller damit werben, dass sie sich an derzeit gültige Standards und Normen halten und somit für eine Investitionssicherheit stehen, stellt sich nach einer Datenmigration die Frage, ob der umge-

setzte Datenbestand sich in der Praxis wirklich so standardisiert verhält, wie es ursprünglich geplant und angekündigt wurde. Wenn man sich also bei einer Datenmigration an die von den GIS-Softwareherstellern unterstützten Datenbanksysteme und deren Datenmodellierung hält, ist es dann möglich, systemunabhängig auf Datenbestände zuzugreifen? Kann man vielleicht sogar schon von „Interoperabilität“ sprechen?

Oder kurz gesagt: „Datenmigration – alles Standard?“

Am Beispiel eines konkreten Anwendungsfalls bei der Firma Dynamit Nobel GmbH zeigt die vorliegende Arbeit die Problematik und die Lösungsansätze einer Datenmigration. Es wird gezeigt, mit welchen Methoden eine Datenmigration innerhalb verschiedener GIS-Systeme nach heutigen Gesichtspunkten durchführbar ist. Abschließend wird erläutert, inwiefern eine systemübergreifende Nutzung des migrierten Datenbestandes realisierbar ist.

2 Ausgangssituation

In diesem Kapitel wird darauf eingegangen, wie sich die allgemeine und die datenverarbeitungstechnische Situation des exemplarisch gewählten Unternehmens in der Vergangenheit darstellte. Anschließend werden die daraus entstandenen Ziele der Datenmigration aufgegriffen und erläutert.

2.1 Firma Dynamit Nobel

Die Entstehung der heutigen Firma Dynamit Nobel geht zurück bis in das Jahr 1865, in dem der Schwede Alfred Nobel mit der Firmengründung in Hamburg den Grundstein legte. In dieser Zeit, die als Beginn der Industrialisierung in Deutschland gilt, wurde der Bedarf an Sprengstoffen immer größer. Nachdem er zwei Jahre später das Dynamit erfunden hatte, wurde 1874 im Rheinland das zweite Sprengstoffwerk in Deutschland errichtet. Zu diesem Zeitpunkt ließ sich Nobel seine sehr erfolgreiche Erfindung schon in vielen Ländern patentieren. Bereits zwölf Jahre später wurde mit dem Bau eines weiteren Werkes in Troisdorf begonnen, wo im Jahre 1905 der erste technisch verwertbare Kunststoff produziert wurde. Dabei verarbeitete man den Sprengstoffrohstoff Nitrocellulose zu Celluloid. Als sich Deutschland zu Beginn der fünfziger Jahre im Wiederaufbau befand, stieg die Firma mit der Entwicklung organischer Zwischensubstanzen in die Chemie ein. Daraus resultierend wurde im Jahr 1970 im Rheinland mit dem Aufbau der Spezialchemie begonnen.

Ende der achtziger Jahre konnte der Dynamit Nobel Konzern einen Umsatz von über 1,5 Milliarden Euro vorweisen. Etwa ein Viertel davon entfiel dabei auf die traditionelle Sprengmittelherstellung. Den weitaus größeren Teil bildeten die chemischen Grundstoffe und die Kunststoffverarbeitung.

Anfang der neunziger Jahre erwarb die Metallgesellschaft AG die Aktienmehrheit an der Firma. In der Folge gab es viele verschiedene Fusionierungen und Neugliederungen, von denen mehr oder weniger Teilbereiche der Firma Dynamit Nobel betroffen waren.

Heute ist die Firma ein weltweit operierendes Unternehmen mit fast 15.000 Mitarbeitern und einem Umsatz von über 2,6 Milliarden Euro. Sie gilt als Spezialist für hochwertige Produktionen in den Bereichen Chemie und Werkstofftechnik.

Die heutigen Geschäftsfelder Sprengmittel, Kunststoffe, Hochleistungskeramik, Spezialitätenchemie und Pigmentchemie werden von operativ tätigen Tochtergesellschaften geführt. Der Hauptsitz der Dynamit Nobel AG ist in Troisdorf, von wo aus der Konzernverbund als strategische Managementholding geführt wird (Dynamit Nobel GmbH, 2002).

In den 80er Jahren wurde damit begonnen, einen Datenbestand auf digitaler Basis aufzubauen. Dabei wurden zunächst CAD-Pläne aus den vorhandenen analogen Aufzeichnungen erstellt. Diese Pläne verbunden mit Informationsmaterialien in Aktenform bildeten einen immerhin beachtlichen Datenbestand, der in den folgenden Jahren stets fortgeführt und erweitert wurde. Im Jahr 1990 wurde die Software Topodat von CPlan eingeführt, um den vorhandenen Datenbestand dort einzupflegen und spätere Vermessungen und Planungen auf digitalem Wege festzuhalten und zu realisieren. Diese Software basierte damals auf einem Unix-System und wurde 1997 durch eine Nachfolgerversion, die auf einer Windows-Plattform basierte, abgelöst. Diese Software soll im nächsten Abschnitt hinsichtlich der Datenhaltung kurz beschrieben werden, da sie der unmittelbare Vorgänger der derzeitigen Software ist und somit als der zu migrierende Datenbestand galt.

2.2 Software Topodat von C-Plan

Die topographische Datenbank Topodat besteht aus einem Grundmodul und verschiedenen optionalen Ergänzungen, die je nach Anwendungsbereich hinzugefügt werden können. Solche Zusatzmodule sind beispielsweise das Modul Vermessung, das Modul Kanal, oder Module zur Berechnung und Erstellung von Quer- und Regelprofilen im Strassen- oder Wasserbau. Topodat eignet sich sehr gut für den Einsatz in den Bereichen der Vermessung und Planung, in denen heute nach wie vor noch vielfach auf diese Lösung gesetzt wird.

Die Datenablage ist keineswegs standardisiert, sondern erfolgt in einem eigenen, proprietären Datenformat und ist stets punktbezogen. Die Daten werden in einzelnen Dateien abgelegt, die die Endung *.top haben.

Die Hauptdatei stellt die „elemda.top“ dar. Diese Datei ist eine binäre Indexdatei, also kein ASCII-Format. Allerdings ist für jeden Punkt ähnlich wie in einer ASCII-Datei eine Datenzeile vorhanden.

Wenn man mit einem beiliegendem Compiler die Datei elemda.top als ASCII-Datei darstellt, stellt sich folgende Struktur dar:

Element	Positionen
Punktnummer	6 – 15
Vermarkungsart	28 – 30
Anderungsnummer	40 – 45
Bestimmungsklasse	60
X-Koordinate	67 – 75
Y-Koordinate	82 – 90
Höhe	99 – 105
Orientierung	112 – 120
Y-Handriß	129 – 135
X-Handriß	144 – 150
Y-Schriftpos. Punktnr.	157 - 165
X-Schriftpos. Punktnr.	172 – 180
Richtung der Punktnr.	189 – 195
Richtung Symbol	209 – 215
Datum	238 – 239
Mutationsnummer	252 – 260
Entstehungscode	275
Gütecode	290
Genauigkeitsstufe	305
Ebene 1	321 – 325
Gruppe	401 – 405
Liniendefinitionsnr.	412 – 420
Von Punktnummer	427 – 435
Nach Punktnummer	442 – 450
Darstellungsart	463 – 465
Veränderungscode	479 – 480
Lagegenauigkeit	487 – 495
Aufnahmedistanz	502 – 510
Aufnahmerichtung	517 – 525
Hist 1 – Hist 8	530 – 645
Radius	650 – 660
Typ	670

Tabelle 2.1: Struktur der elemda.top

Weiterhin gibt es die Tabellen „punu.top“, „punam.top“, „linien.top“, „beras0.top“ und „pusu0.top“.

Die Datei punu.top ist eine Indexdatei auf die Punktnummern. Wenn in der elemda.top ein Elementtyp „Punkt“ abgelegt ist, gibt es zu diesem auch eine Punktnummer. In der punu.top ist angegeben, unter welcher Elementnummer ein Punkt in der elemda.top zu finden ist.

Neben der Punktnummer kann auch ein Punktname in Form eines Textes in der elemda.top mit abgelegt werden. Daher bildet die Datei punam.top eine Indexdatei auf den Punktnamen, also einen Index auf den Sekundärnamen eines Punktes.

Die Datei linien.top stellt eine Indexdatei für die Linien dar. In ihr wird eine Linie durch die zugehörigen Punkte definiert. Wenn der Anfangs- und Endpunkt einer Linie identisch ist, die Linie also in sich zusammenläuft, dann bildet diese Linie den Umring einer Fläche.

Ähnlich der heutigen räumlichen Indizierung von Elementen gibt es in der Topodat die Möglichkeit, ein Raster über den Datenbestand zu legen. In der Datei beras0.top wird eine Kachelung von 100m x 100m für alle Elemente definiert. Diese Kachelgröße ist fixed und dient in erster Linie einem schnelleren Bildaufbau aus der Topodat heraus.

Die Datei pusu0.top bildet ebenfalls eine solche Rasterung, jedoch nur auf die Punkte bezogen, nicht etwa auf die Texte.

Weiterhin gibt es noch sogenannte Macrodateien, die die Endung *.mac haben. In diesen Dateien sind eine Vielzahl von Symbolen definiert, die entweder vorgegeben sind, oder auch vom Benutzer erstellt und angepasst werden können. Diese Symbole haben wiederum einen definierten Punkt als Aufhängepunkt (C-Plan (a), 1999)

Da die einzelnen Standorte des Konzerns auf ihren Firmenarealen umfangreiche Kanalnetze betreiben, ist der Betreiber schon von Gesetzes Wegen dazu verpflichtet, die Abwasserwege zu überwachen und stets Buch über deren Zustand zu führen. Auch diese Aufgaben werden unter Zuhilfenahme eines Informationssystems bewerkstelligt. Dieses System stellte in der Vergangenheit die Software „CINS plus“ von C-Plan dar. Dabei handelt es sich um eine Kanaldatenbank, die auf der Basis von Access aufgebaut ist. In dieser Datenbank gibt es Tabellen für Schächte, Haltungen, Inspektionen, Sonderbauwerke oder Zustände der Bauteile. Über diese entsprechend miteinander verbundenen Tabel-

len ist ein Menü programmiert, über das der Nutzer auch ohne weitere Access-Kenntnisse das Programm bedienen kann. Über den Haltungs- bzw. den Schachtnamen gibt es eine Verknüpfung zur Topodat, die dann als Grafikanbindung dient. Wenn beide Programme gestartet sind, kann sich daher die in CINS plus ausgewählten Elemente über die elemda.top in der Grafik anzeigen lassen.

Der folgende Screenshot zeigt eine aus der Topodat erzeugte Gesamtansicht eines Werksgeländes. Zusätzlich wurde ein Planrahmen, Schriftfeld und eine Legende angelegt, so dass eine komplett ausgearbeitete Karte entstanden ist.

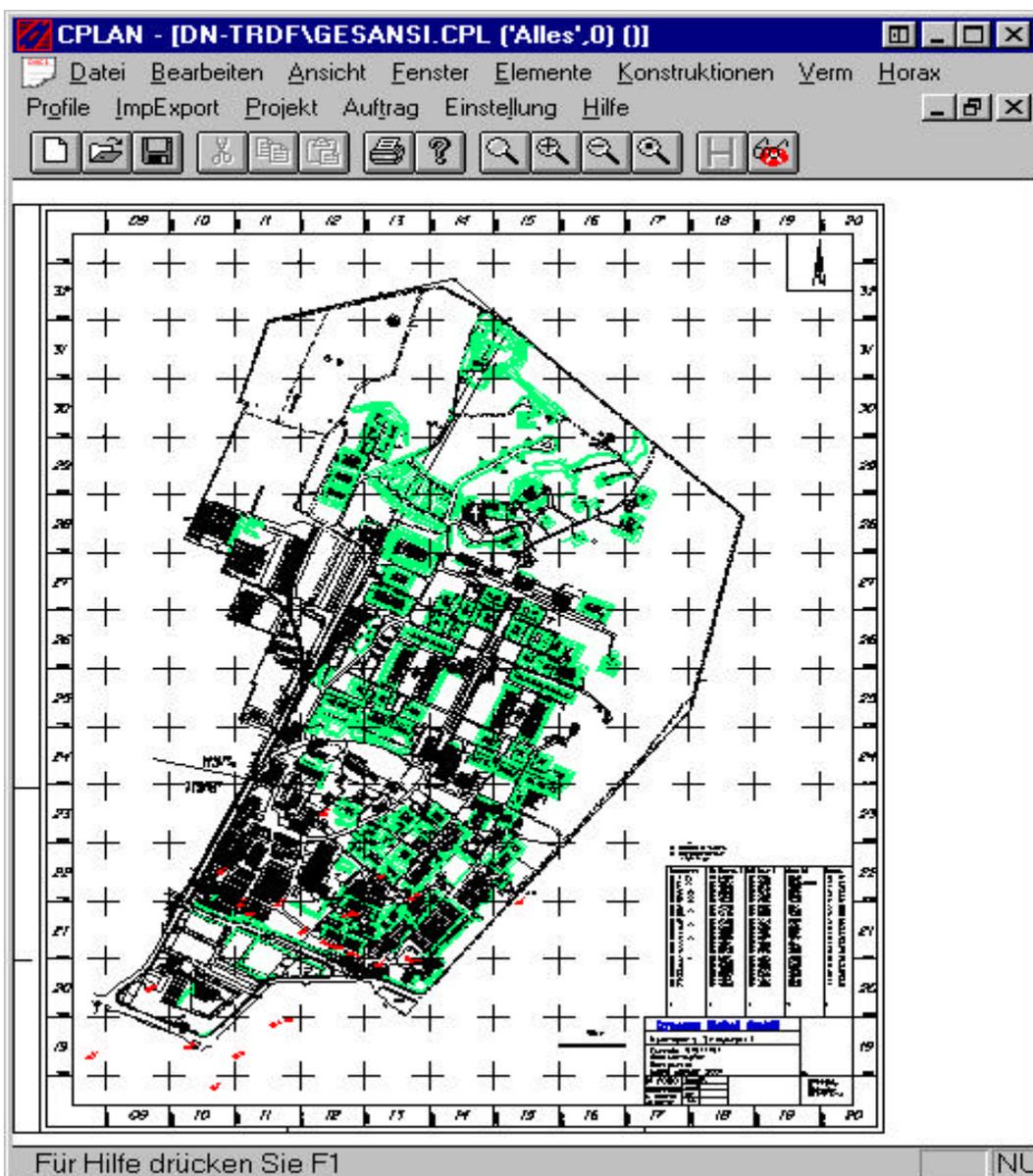


Abbildung 2-1: Karte aus Topodat

2.3 Zielsetzung

Der Wunsch, den über Jahre gewachsenen Datenbestand in ein standardisiertes System zu überführen, ist aus verschiedenen Ursprüngen heraus entstanden, die letztendlich dazu geführt haben, eine Datenmigration anzugehen. Im Wesentlichen lassen sich die Erfahrungen aus der Vergangenheit und die Wünsche für die Zukunft auf die beiden Überschriften reduzieren, die Einrichtung von Auskunftsarbeitsplätzen als originärer Nutzen der Daten und die Schaffung einer offenen Umgebung für den Datenbestand, so dass auch dritte Systeme auf diesen Datenbestand zugreifen können.

In der Vergangenheit leistete sich das Unternehmen noch eine eigene Bauabteilung, die in Ihrer Zuständigkeit die verschiedenen Standorte des Konzerns bezüglich Planung und Durchführung von baulichen Maßnahmen betreute. Diese Abteilung bestand aus Ingenieuren und Technikern, die u.a. den Datenbestand aufgebaut und gepflegt haben. Im Zuge von Rationalisierungsmaßnahmen und betriebsbedingten Einsparungen wurde diese Abteilung im Laufe der Zeit aufgelöst. Die Datenpflege wurde auf eine Person reduziert, wobei die Datenerfassung durch Vermessung und auch planerische Aufgaben weitestgehend an Fremdfirmen vergeben wurde. Nunmehr war eine Person dafür zuständig, alle Standorte mit Karten- und Sachdaten zu versorgen. Dies geschah meistens in Form von analogen Plänen, analogen Ausdrucken als Ergebnis von geometrischen oder zeitlichen Analysen, oder auch verbalen Auskünften. Später hat ein mittelständiges Bauingenieurbüro die Verwaltung des Systems übernommen. Dieses Büro lag geographisch etwa 80 km zwischen zwei Standorten. Von dort aus wurden die selben Leistungen erbracht, wie zuvor von der einen Person, die an einem Standort beschäftigt war. Der Vorteil dieser Konstellation war, dass nun Hochbau-, Tiefbau- und GIS-Abteilungen verschiedene Planungsaufgaben übernehmen konnten. Der Konzern hatte nun die selben Verhältnisse wie bereits vor einiger Zeit geschaffen, nur dass es jetzt keine eigene Bauabteilung mehr gab, sondern ein externes Ingenieurbüro mit diesen Aufgaben betraut wurde.

Zwar funktionierte der Informationsaustausch zwischen Ingenieurbüro und Konzern sehr gut, jedoch war es stets mit einem gewissen Aufwand verbunden, wegen jeder kleinen Auskunft über das Telefon oder per Email das Büro zu

bemühen. Es kam daher sehr schnell der Wunsch auf, wenigstens einen gewissen Teil an Informationen, der über eine analoge Karte hinaus ging, stets vor Ort präsent zu haben. Allerdings waren die Ansprüche an solche Informationen und Auskünfte zwischen den zuständigen Mitarbeitern des Konzerns sehr verschieden. So wollte der Beauftragte für die Werkselektrik stets einen aktuellen Plan über die Verläufe von Stromleitungen vorhalten, während der Sicherheitsbeauftragte mit ganz anderen Fragestellungen konfrontiert wurde. Der Abwasserbeauftragte musste stets – auch dem Gesetzgeber gegenüber – Auskunft über die Verläufe und den Zustand des Kanalnetzes geben. Auf dieser Basis mussten Sanierungskonzepte und Kostenschätzungen gemacht werden.

Alle diese verschiedenen Fragestellungen liefen letztendlich auf den selben Datenbestand zurück und konnten aus verschiedensten Zusammenhängen heraus beantwortet werden. Daher entwickelte sich die Frage, ob man nicht von den verschiedenen Standorten aus mit unterschiedlichen Prioritäten online auf den Datenbestand zugreifen könnte, um sich nur die Informationen ausgeben zu lassen, die für die jeweilige Abteilung von Interesse ist. Da ein solches Vorhaben heute mit Hilfe der Web-Technologie technisch kein Problem mehr darstellt, war zu klären, inwiefern der Datenbestand sich dazu eignete, die Informationen netzwerkartig zu verteilen. Zu diesem Zeitpunkt hatte bereits die Firma C-Plan die Nachfolgesoftware der Topodat, die Topobase, entwickelt und auf den Markt gebracht. Diese neue Lösung sieht auch entsprechende Module zur Informationsverteilung mit Hilfe der Web-Technologie vor. Die Software Topodat wurde nach dem Erscheinen der Topobase von C-Plan nicht weiter entwickelt, sondern nur noch supported. Eine Weiterentwicklung in Richtung Internet und verteilte Informationen war nicht angedacht.

Darüber hinaus eignete sich die vorhandene Art der Datenhaltung in der Topodat in keiner Weise dazu, ein Web-basiertes Auskunftssystem aufzubauen. Die Topodat wurde Mitte der 80er Jahre entwickelt und diente stets nur der Verwendung als Einzelarbeitsplatz und wurde nicht für den Client/Server-Betrieb konzipiert.

Somit war ein Aspekt gegeben, mit in die neue Softwaregeneration von C-Plan zu gehen, oder sich zumindest generell mit dem Gedanken einer Datenmigration auseinander zu setzen.

Durch die Veröffentlichung der neuen Software von C-Plan Topobase und der damit verbundenen Einstellung der Weiterentwicklung der Topodat sah sich der Konzern bezogen auf den Datenbestand schnell in der Gefahr, mit dem vorhandenen System, auf längere Sicht betrachtet, kaum handlungsfähig zu sein. Zwar konnten mit der Topodat alle Aufgaben, die bisher bewältigt werden mussten auch weiterhin bearbeitet werden, aber es wurden keine weiteren Updates oder Bugfixes mehr seitens C-Plan angeboten. Es wurde die Situation deutlich, dass man sich mit diesem System eine sogenannte Insellösung geschaffen hat. Man konnte weder fremde Daten zuspülen, noch war es möglich, die Topodat-Daten anderweitig – abgesehen von ASCII und DWG-Zeichnungen – zu verwenden.

Das Problem der Insellösung wurde auch in der Vergangenheit bereits mehrfach deutlich, als es hilfreich gewesen wäre, wenn man sich mit anderen GI- oder CAD-Systemen hätte austauschen können. Daher war es oft umständlich einen Datentransfer zwischen dem Konzern und den Behörden, oder Planungsbüros zu gewährleisten.

Weiterhin kam der Wunsch auf, eine Datenhaltung zu führen, die sich nicht unbedingt auf ein spezielles System bezieht, sondern offen ist für eine Anbindung verschiedener Fachlösungen. Diese Vorstellung resultierte daraus, dass man sich nun ungern auf ein neues System stützen wollte, welches vielleicht in drei oder fünf Jahren wieder eine Insellösung darstellen würde.

Es hat sich gezeigt, dass obwohl sich niemand der Konzernmitarbeiter mit geographischen Informationssystemen tiefer beschäftigt hat, trotzdem die Wünsche und Vorstellung unter den Begriffen „offene Datenhaltung“ und „verteilte Informationen“ wieder zu finden waren.

Zwar bietet die Software Topodat sehr umfangreiche Funktionalitäten im Bereich der Vermessung und Konstruktion an, jedoch sind kaum GIS-typische Analyse- und Abfragefunktionalitäten vorhanden. Da der Konzern in jüngster Zeit in großem Umfang umstrukturiert wurde, kamen aufgrund neuer Besitz- und Abrechnungsverhältnisse ganz neue Fragestellungen auf, deren Beantwortung man unter heutigen Gesichtspunkten durchaus zu den Grundfunktionalitäten einer GIS-Software zählen kann. Solche Fragestellungen können beispielsweise sein: „Wieviel Kilometer Strasse, oder Kanal verlaufen auf dem Gebiet eines bestimmten Eigentümers?“, oder „Wie groß ist die Gesamtfläche aller Büroge-

bäude eines Standortes?“. Auch sicherheitstechnische Fragestellungen, wie z.B.: „Welche Gebäude liegen innerhalb von 100m von einer Produktionsstätte entfernt und wären bei einer Explosion betroffen?“ sind von Belange. Solche GIS-typischen Analyse- und Abfragefunktionen konnten aber mit der Topodat-Software nicht, oder nur sehr umständlich gelöst werden, so dass auch auf diesem Gebiet der Wunsch nach einer Neuerung seitens derer, die mit dem System gearbeitet haben, lauter wurde.

2.4 Zusammenfassung

Das vorgestellte Industrieunternehmen Dynamit Nobel GmbH ist ein Konzern der sich auf die Herstellung und Verarbeitung von Sprengstoffen und Kunststoffen spezialisiert hat. Gewachsen aus historischen Anfängen steht der Name heute für ein weltweitoperierendes Unternehmen mit komplexen Strukturen. Mit dem Hauptsitz in Troisdorf/NRW setzte Dynamit Nobel in der Vergangenheit auf die Softwarelösungen der Schweizer Firma C-Plan. Auf der Basis des Produktes Topodat wurden Anfang der 90er Jahre vorhandene analoge Pläne übernommen und fortan stetig neue Daten durch Vermessungen und Begehungen erfasst. Die Software Topodat versteht sich als topographische Datenbank. Durch einem modularen Aufbau können verschiedenste Fachdisziplinen angesprochen und bearbeitet werden. Die Datenablage erfolgt in einem proprietären Dateiformat, wobei in der Hauptdatei elemda.top alle Informationen abgelegt werden. Darüber hinaus gibt es noch weitere Dateien, die meistens Indexdateien darstellen. Aus Gründen wie Investitionssicherheit, mangelnde Flexibilität bei der Verteilung der Daten, fehlende Analyse- und Abfragefunktionalitäten ergaben sich im Laufe der Zeit Wünsche und Ziele, die mit einem Systemwechsel und einer damit verbundenen Datenmigration realisiert werden sollten. Die beiden Hauptziele, Schaffung von Auskunftsarbeitsplätzen als originäres Ziel und das Erreichen einer investitionssicheren und offenen Datenhaltung, standen dabei im Vordergrund.

3 GIS-Architektur des Projektes

Die modernen und derzeit aktuellen GIS-Lösungen bauen bei der Konzeption des GIS-Komponentengefüges auf die sogenannte 3-tier- (3-Schichten)-Architektur. In einer solchen Anordnung können verschiedenste Komponenten eingebettet werden. Die drei Schichten stellen sich wie folgt dar.

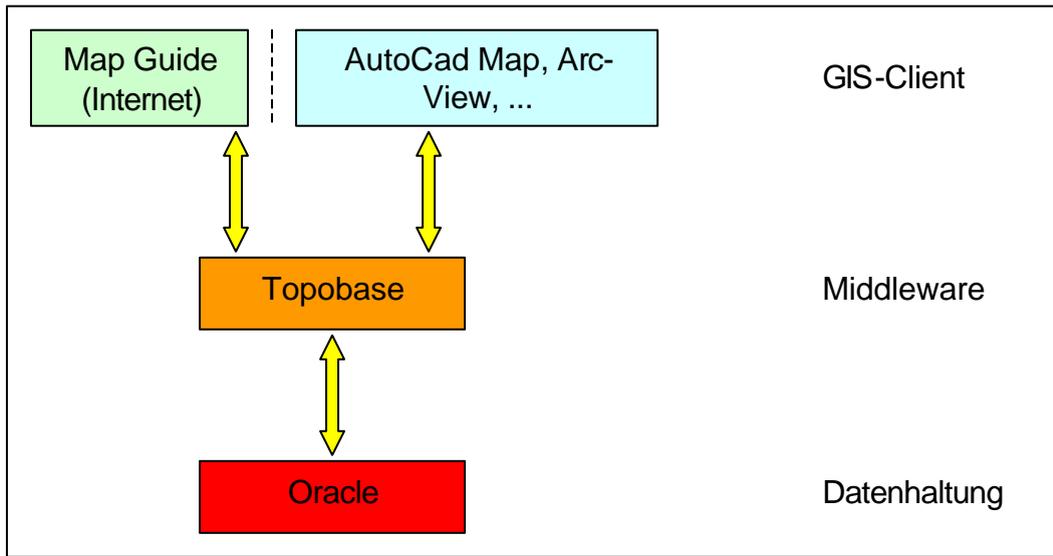


Abbildung 3-1: 3-tier-Architektur

Durch eine softwaretechnische Trennung dieser Schichten ist es möglich, verteilbare und skalierbare Lösungen zu schaffen.

3.1 Datenhaltung der Spatial-Architektur

Die Ende der siebziger Jahre gegründete Firma Oracle ist heute in Redwood Shores, Kalifornien beheimatet und bietet Softwareprodukte und Dienstleistungen für das elektronische Management von Informationen an. Mit dem DBMS Oracle 8i, welches 1997 erstmals auf dem Markt erschien, bietet die Firma eine leistungs- und voll internetfähige Enterprise Software an. Dazu zählen Datenbanken, Server und Enterprise Business Applikationen (Oracle, 2000).

Oracle 8i ist das erste Datenbanksystem, welches über die Datenbankfunktionalität hinaus auch eine umfangreiche Plattform zur Entwicklung und Einsatz von internetbasierten Anwendungen bietet. Das System verfügt über eine im Datenbank-Kernel integrierte Java Virtual Maschine (Oracle Jserver Enterprise Edition), die Java Anwendungen schneller

Edition), die Java Anwendungen schneller und effizienter auf Oracle unterstützten Plattformen ablaufen lässt. Mit der Version 8i wurden die klassischen Merkmale Datenbankmanagement, Transaktionsverarbeitung und Data Warehouse um die Komponente Internet erweitert.

Die Enterprise Edition zeichnet sich durch die entsprechenden Administrationswerkzeuge und Skalierbarkeit aus, wodurch eine optimale Performance erreicht wird und eine Verteilung der Informationen im Internet oder Intranet problemlos möglich ist. Es werden hochvolumige OLTP-Anwendungen und größere Data Warehouse-Systeme unterstützt.

Für bestimmte Anwendungsgebiete bietet Oracle für das DBMS Erweiterungen an, die in ihren Inhalten weit über die standardmäßige Zusammensetzung hinaus gehen. Diese Module enthalten Funktionalitäten, Prozeduren und Objekttypen, die speziell für diesen Anwendungstyp definiert wurden. Solche Erweiterungen werden Cartridges genannt.

Exemplarische Beispiele für solche Module sind die Image-Cartridge, die Context-Cartridge oder die Spatial-Cartridge.

Im Zuge der vorliegenden Arbeit ist die Spatial-Cartridge von besonderer Bedeutung, da die Middleware Topobase diese zur Datenspeicherung verwendet. Daher soll sie im folgenden etwas näher beschrieben werden.

Mit der Spatial Cartridge wird das relationale Datenbankmanagementsystem Oracle um eine räumliche Komponente erweitert. Man spricht dabei von einem „Spatial Enabled DBMS“. Dadurch ist es möglich, räumliche Daten ebenso in einem DBMS abzulegen, wie herkömmliche, nichträumliche Daten. Die Spatial Cartridge bietet dabei weiterhin die Funktionalitäten und Prozeduren, um die Speicherung, Abfragen, Verwaltung und Fortführung räumlicher Daten zu realisieren. Dazu stellt die Erweiterung folgende Komponenten zur Verfügung:

- Räumliche Analysefunktionen und –prozeduren
- Räumliche Abfrageoperatoren und –funktionen
- Räumliche Indizierung
- Ein vordefiniertes Schema, das die Speicherung, die Syntax und die Semantik der unterstützten geometrischen Datentypen beinhaltet (MDSYS)
- Administrationswerkzeuge

Im folgenden soll auf einige Grundlagen und Funktionalitäten von Oracle Spatial 8i eingegangen werden, da diese im weiteren Verlauf der Projektarbeit verschiedentlich Verwendung finden.

3.1.1 Modellierung

Bei der Betrachtung der Modellierung muss zunächst unterschieden werden zwischen dem logischen und dem physischen Datenmodell.

Das logische Datenmodell von Oracle Spatial besteht aus Elementen, Geometrien und Layern, die in einer Hierarchie angeordnet sind.

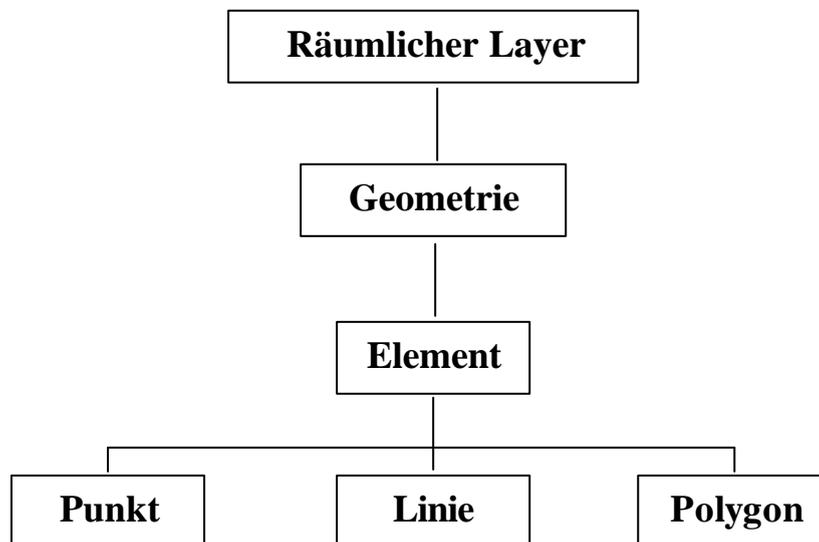


Abbildung 3-2: Logische Datenmodell von Oracle

Aus den geometrischen Primitiven wie Punkt, Linie, Polygon oder auch komplexere Formen werden die Elemente aufgebaut. Diese Elemente bilden die Geometrien, welche eine Menge von Attributen haben können. Eine oder mehrere Geometrien sind einem räumlichen Layer zugeordnet, welchem ein definierter Objektraum und ein Koordinatensystem zugeteilt ist.

Zur Speicherung von räumlichen Daten werden in Oracle Spatial zwei physische Datenmodelle unterstützt.

Das relationale Modell: Das geometrische Objekt wird in einer Tabelle mit einer vordefinierten Anzahl von Spalten vom Typ Number abgelegt.

Das objektrelationale Modell: Die Geometrie wird in einer einzigen Spalte vom objekt-relationalen Datentyp MDSYS.SDO_Geometry gespeichert.

Oracle Spatial unterstützt im relationalen Modell drei zweidimensionale Geometrietypen, die man als die geometrischen Primitiven bezeichnet. Dies sind Punkte, Linien/Linienzüge und Polygone. Weiters werden Kollektionen dieser Typen unterstützt.

Im objekt-relationalen Modell sind darüber hinaus noch folgende Geometrietypen enthalten: Kreisbogenfolgen, Polygone aus Kreisbögen, zusammengesetzte Polygone, zusammengesetzte Linienzüge, Kreise und optimierte Rechtecke. Auch hierbei werden wieder Kollektionen dieser verschiedenen Typen unterstützt, wobei alle Geometrietypen aus Koordinatenpaaren bestehen, die in beliebigen Koordinatensystemen vorliegen können.

Die beiden Modelle gehen konform mit den OpenGis SimpleFeaturesSpecifications for SQL, wobei bei OGC nur die drei Primitiven als Geometrietyp definiert sind.

Im relationalen Objektmodell von Oracle Spatial werden die Daten und deren Indizierung in vier Tabellen abgelegt. Diese Datenbankstruktur entspricht der ersten von drei OpenGIS Features for SQL Implementation Options. Es werden numerische SQL Statements zur Speicherung der Geometrien benutzt.

Diese vier Tabellen beziehen sich jeweils auf einen Layer, werden also pro Ebene angelegt. Für die Erstellung der Tabellen wird ein Vorlage-SQL-Script bereitgestellt (crlayer.sql).

Die folgenden Tabellen beschreiben das Schema eines räumlichen Layer „Linien“:

Linien_SDOLAYER

SDO_ORDCNT	SDO_LEVEL	SDO_NUMTILES	SDO_MAXLEVEL	SDO_COORDSYS
<number>	<number>	<number>	<number>	<varchar>

Tabelle 3.1: Linien_SDOLAYER

(SDO_MAXLEVEL und SDO_COORDSYS sind optionale Spalten)

- SDO_ORDCNT: Diese Spalte beinhaltet die Gesamtanzahl der Ordinaten pro Zeile der Tabelle Linien_SDOGEOM, also die Anzahl der Spalten, die Datenwerte enthalten.
- SDO_LEVEL: Dort wird der Teilungsgrad der Tessellation abgelegt.
- SDO_NUMTILES: Diese Spalte wird Null gesetzt, wenn bei der Teilung eine feste Kachelgröße verwendet wird. Ansonsten wird dort die Anzahl der Kacheln abgelegt.
- SDO_MAXLEVEL: In dieser Spalte wird der höchste Teilungsgrad eingetragen, wenn die Kachelgröße als variabel gewählt wird.
- SDO_COORDSYS: Es kann der Name des Koordinatensystems abgelegt werden, wie z.B. POSC oder OGIS.

Linien_SDODIM

SDO_DIMNUM	SDO_LB	SDO_UB	SDO_TOLERANCE	SDO_DIMNAME
<number>	<number>	<number>	<number>	<varchar>

Tabelle 3.2: Linien_SDODIM

- SDO_DIMNUM: In dieser Spalte wird die Dimension abgelegt, auf welche sich diese Zeile bezieht. Die Werte beginnen mit 1 und steigen an.
- SDO_LB: Diese Spalte beschreibt die untere Grenze der Ordinate. Dieser Wert ist z.B. -90, wenn die Dimension die Breite ist.
- SDO_UB: Diese Spalte beschreibt die obere Grenze der Ordinate. Wenn die Dimension wieder die Breite ist, wird dieser Wert 90 annehmen.
- SDO_TOLERANCE: Die Toleranz stellt den Abstand dar, die zwei Punkte voneinander haben können. Dieser Wert muss größer Null betragen.
- SDO_DIMNAME: In dieser Spalte wird der Name der Dimension abgelegt, also z.B. longitude, latidute, X oder Y.

Linien_SDOGEOM

SDO_GID	SDO_ESEQ	SDO_ETYPE	SDO_SEQ	SDO_X1	SDO_Y1
<number>	<number>	<number>	<number>	<number>	<number>

Tabelle 3.3: Linien_SDOGEOM

(zzgl. SDO_Xn und SDO_Yn für weitere Koordinatenpaare)

- SDO_GID: In dieser Spalte wird ein eindeutiger, numerischer Wert für jede Geometrie eines Layers abgelegt.
- SDO_ESEQ: Diese Spalte zählt jedes Element einer Geometrie auf. Dieser Wert wird als Element SEQUENCE number bezeichnet.
- SDO_ETYPE: In dieser Spalte wird der Geometrietyp gespeichert. Die gültigen Werte lauten SDO_GEOM.POINT_TYPE, SDO_GEOM.LINestring_TYPE, SDO_GEOM.POLYGON_TYPE (1, 2 oder 3).
- SDO_SEQ: Diese Spalte speichert die Sequenz-Nummer jeder Zeile, die das Element beschreibt.
- SDO_X1: Stellt den X-Wert der ersten Koordinate dar.
- SDO_Y1: Stellt den Y-Wert der ersten Koordinate dar.

Linien_SDOINDEX

SDO_GID	SDO_CODE	SDO_MAXCODE	SDO_GROUPCODE	SDO_META
<number>	<raw>	<raw>	<raw>	<raw>

Tabelle 3.4: Linien_SDOINDEX

(SDO_MAXCODE und SDO_GROUPCODE werden nicht benötigt, wenn man bei der Indizierung die empfohlene fixe Größe der Kacheln verwendet.)

- SDO_GID: siehe oben
- SDO_CODE: In dieser Spalte ist die Kachel-ID einer Teilüberlappung, die die SDO_GID bedeckt, enthalten.
- SDO_MAXCODE: Dieser Code beschreibt die kleinste Kachel eines Quadranten, wenn die Kachelgröße variabel ist.
- SDO_GROUPCODE: Dieser Code ist ein Präfix des SDO_CODE.

- SDO_META: Diese Spalte ist für räumliche Abfragen nicht erforderlich. Sie enthält Informationen, die benötigt werden, um die Grenzen einer Kachel zu bestimmen.

Der grundlegende Unterschied vom objektrelationalen Modell zum relationalen Modell liegt darin, dass beim objektrelationalen Ansatz die Geometrie in einer einzigen Spalte gespeichert wird. Diese Spalte hat den objektrelationalen Datentyp SDO_GEOMETRY.

Dieser Objekttyp wird in Oracle Spatial folgendermaßen definiert:

```
Create Type SDO_GEOMETRY As Object (  
    SDO_GTYPE      Number,  
    SDO_SRID       Number,  
    SDO_POINT      MDSYS.SDO_POINT_TYPE,  
    SDO_ELEM_INFO  MDSYS.SDO_ELEM_INFO_ARRAY,  
    SDO_ORDINATES  MDSYS.SDO_ORDINATE_ARRAY);
```

- SDO_GTYPE: Durch dieses Attribut wird der Geometriertyp festgelegt. Oracle unterstützt hier sieben Geometriertypen.
- SDO_SRID: In diesem Eintrag wird ein Schlüssel des zugrundeliegenden Koordinatensystems eingebracht.
- SDO_POINT: In diesem Feld wird ein Objekt vom Objekttyp MDSYS.SDO_POINT_TYPE mit den Attributen X, Y und Z vom Typ Number definiert. Wenn die beiden verbleibenden Felder SDO_ELEM_INFO und SDO_ORDINATES leer bleiben, wird das Objekt als Punktobjekt mit den Koordinaten X, Y und Z interpretiert. Sind die Felder hingegen ausgefüllt, so wird der Eintrag in SDO_POINT ignoriert.
- SDO_ELEM_INFO: Hinter dieser Variablen verbirgt sich das systemdefinierte Listenfeld SDO_ELEM_INFO_ARRAY. Dieses Feld basiert auf dem Objekttyp SDO_ELEM_INFO_TYPE.

SDO_ORDINATES: Dieses Feld beinhaltet die eigentlichen Koordinaten des Objektes. Dabei handelt es sich ebenfalls um ein Listenfeld, aber des Typs SDO_ORDINATE_ARRAY.

3.1.2 Abfragemodell

Um räumliche Daten abzufragen benutzt Oracle Spatial ein Abfragemodell, welches sich auf zwei Ebenen bewegt. Man bezeichnet diese beiden hintereinander ablaufende Operationen Primär- und Sekundärfilter. Bei einer räumlichen Abfrage bildet zunächst der Primärfilter durch geometrische Approximationen einen kleinen Kandidatensatz. Dabei kommt der räumliche Index zum Tragen. Diese Auswahl wird dem Sekundärfilter zugeführt. Aus dem Auswahlsatz ermittelt der Sekundärfilter dann durch implementierte räumliche Funktionen das Ergebnis der eigentlichen Abfrage. Zwar ist der Sekundärfilter sehr komplex, jedoch wird durch das Vorschalten des Primärfilters die Filterfunktion stets nur auf einen Datensatz geringer Größe angewendet. Dieses Verfahren bietet eine sehr gute Skalierbarkeit bei sich vergrößernden Datenmengen. Zudem wird die Verarbeitungsgeschwindigkeit von räumlichen Abfragen wesentlich verbessert.

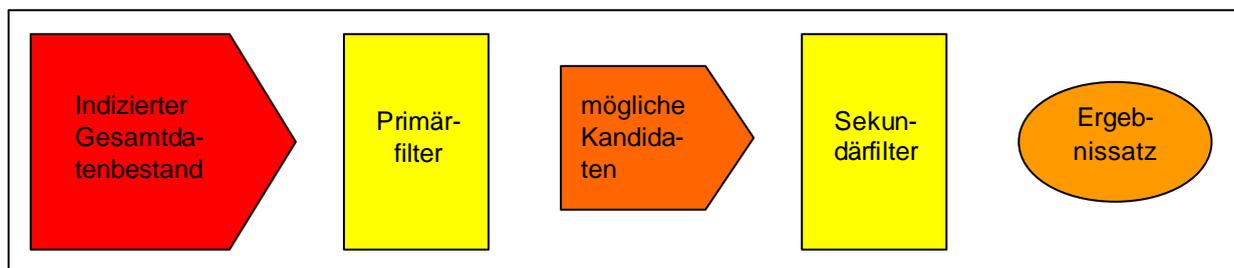


Abbildung 3-3: Räumliche Abfragemodell

3.1.3 Räumliche Indizierung

Eine der wichtigsten Funktionalitäten der Oracle Spatial Cartridge ist die Möglichkeit, die Daten mit einem räumlichen Index zu versehen. Durch diese Indizierung wird die Suche nach Daten in einer Tabelle wesentlich optimiert. Die Bildung des räumlichen Index erfolgt durch eine Prozedur, die Tesselation (engl. Mosaik) genannt wird. Dabei wird das zweidimensionale Koordinatensystem eines Gebietes durch eine lineare Quadtree-Aufteilung mit einer sogenannten Z-Ordnung oder Morton-Ordnung zerlegt, indem pro Teilungsschritt vier weitere Kacheln erzeugt werden. Diese Viertelung wird solange fortgeführt, bis ein Abbruchkriterium erreicht wird. Ein solches Kriterium kann entweder der Teilungsgrad sein (SDO_LEVEL), wobei jede Kachel die gleiche Form und Größe

hat, oder aber auch die Gesamtanzahl der Kacheln (SDO_NUMTILES), wobei die Kachelgröße variabel ist. Das Resultat wird in einer separaten Tabelle abgelegt, die im relationalen Modell <layer>_SDOINDEX heißt. Beim objekt-relationalen Modell vergibt das System den Tabellennamen.

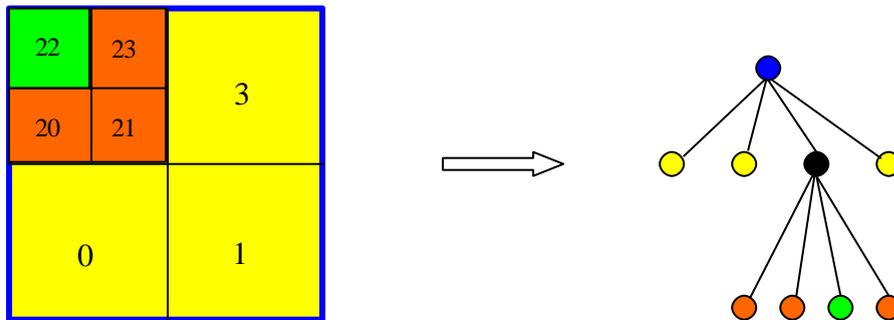


Abbildung 3-4: Quadtrees-Methode

Eine weitere Methode der räumlichen Indizierung ist die des R-Trees. Dabei werden die kleinstmöglichen umschließenden Rechtecke um die jeweiligen Geometrien gebildet und verwaltet. Diese Methode wird allerdings nicht in der Topobase verwendet und daher nicht weiter erläutert.

3.2 Middleware

Die 1996 gegründete Firma C-Plan hat ihren Hauptsitz in Bern (Schweiz) und in Stuttgart (BRD). Darüber hinaus agieren 10 Vertriebspartner, die sich im deutschsprachigen Raum verteilen. Etwa 70 Mitarbeiter, die 2001 einen Umsatz von etwa 7,8 Mio. Euro generierten, zählen sich zu einer der führenden Anbieter von GIS-Lösungen (Buhmann/Wiesel, 2002).

Die Softwarelösung Topobase versteht sich als GIS-Datenbank, bzw. als Geodatenserver, der eine offene und flexible Datenhaltung erlaubt.

Dabei verwendet diese Lösung das Datenbanksystem Oracle mit der Erweiterung Spatial und als Grafikanbindung das GIS AutoCad MapGuide von Autodesk. Geometrie- und Sachdaten werden dadurch in einer einzigen Umgebung gespeichert.

Das folgende Schema zeigt die einzelnen Bestandteile der Topobase-Umgebung, die nachfolgend erläutert werden sollen.

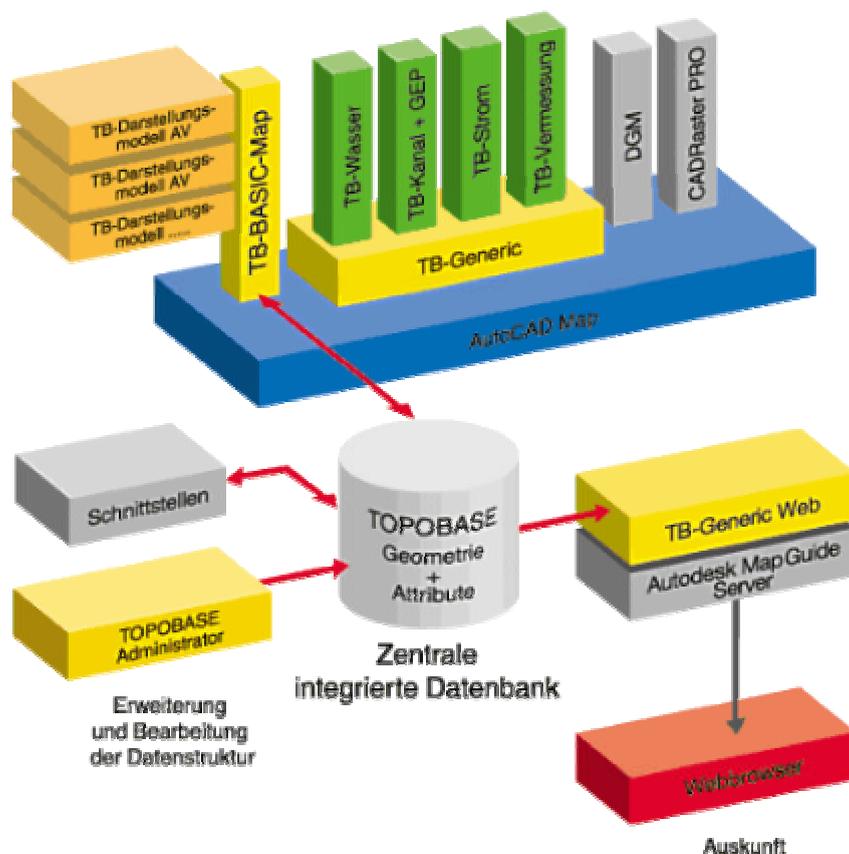


Abbildung 3-5: Bestandteile der Topobase-Umgebung

Topobase: Das Kernstück des Systems bildet die Oracle Datenbank, in der in teilweise vorgefertigten Datenmodellen die Sach- und Geometriedaten abgelegt werden. Vorgefertigt insofern, als dass für verschiedene Anwendungsbereiche bereits beispielhafte Datenmodelle vorhanden sind. Für eigene, bzw. neue Daten können selbstverständlich eigene Datenmodelle zum Einsatz kommen.

TB Administrator: Mit diesem Modul ist es möglich, die Oracle Datenbank zu administrieren. Es werden damit Nutzer angelegt, eine Tabellenstruktur definiert und Felder in die Tabellen eingebracht. In Verbindung mit dem Modul TB_Generic bildet der Administrator ein umfassendes Werkzeug zur Datenmodellierung.

TB Generic: Mit diesem Modul werden anhand des Datenmodells Masken generiert, die der Benutzer beliebig anpassen und gestalten kann. Über diese

Masken, die er in eigenen Menüs verwalten kann, greift der Benutzer auf die Oracle Datenbank zu.

TB Basic Map: Dieses Modul muss zwingend auf jedem Arbeitsplatz installiert sein, an dem geographische Daten erfasst oder bearbeitet werden. Dieses Basismodul ist zwischen der Datenbank und der graphischen Ausgabe anzuordnen. Dort werden die themenbezogenen und maßstabsgerechten Darstellungen festgelegt.

TB Darstellungsmodell: Ein Darstellungsmodell definiert, wie die einzelnen Objekte aus der Datenbank in AutoCad Map gezeichnet werden sollen. Es können für verschiedene Maßstäbe unterschiedliche Darstellungsmodelle definiert werden. Durch Hinterlegen von SQL-Statements können vordefinierte thematische Darstellungen erzeugt werden.

AutoCad Map: Das Standardprodukt von Autodesk dient der Visualisierung der geographischen Daten. Dabei sind alle AutoCad-Funktionalitäten zum konstruieren, bearbeiten, löschen und abfragen von Objekten nutzbar. Durch ein integriertes Menü können die Topobase-Module, wie z.B. TB Basic Map von AutoCad-Map aus aufgerufen werden.

TB Generic Web: Durch dieses Modul ist es möglich, die Daten aus der Oracle-Datenbank im Internet/Intranet zur Verfügung zu stellen. Die Menü- und Maskenstruktur von TB Generic wird nach HTML umgewandelt und ist somit dem Internetnutzer verfügbar.

Autodesk MapGuide Server: Dieses Produkt dient als Applikationsserver bei der Verteilung der Daten über das Internet/Intranet. Es basiert auf dem Internet Information Server von Microsoft und bietet im Zusammenhang mit TB Generic Web die Möglichkeit, die Daten über einen Standardbrowser zu betrachten.

TB Strom: Eines von vielen vordefinierten Anwendungen ist das Modul TB Strom. Für diesen speziellen Anwendungsbereich stehen Datenmodelle, Darstellungsmodelle und technische Funktionalitäten zur Erfassung und Verwaltung von elektrotechnischen Daten zur Verfügung (C-Plan (b), 2002)

In Kapitel 3.1 wurde auf das relationale und das objektrelationale Datenmodell von Oracle Spatial eingegangen. Im folgenden wird beschrieben, wie die Datenablage mit Hilfe der Middleware Topobase realisiert wird.

Um die Daten mit der Software abzulegen, abzufragen und zu analysieren ist es notwendig, dass man unter der Verwendung des TB_Administrators einen Benutzer in der Oracle-Datenbank anlegt. Dabei wird eine von C-Plan entwickelte Struktur in diesem neuen Benutzer erzeugt. Es ist daher nicht möglich, unabhängig von der Software einen Benutzer direkt anzulegen, um mit der Software Daten abzulegen, da dieser dann nicht die Tabellenstruktur aufweist, die C-Plan vorgibt. Es sei denn, man würde mit Hilfe von SQL-Scripts im SQL-Worksheet diese Struktur anlegen. Die Grundstruktur wird zusammen mit den Geometrie- und Sachdaten in einem einzigen Datenbankuser abgelegt, den man dann in der Terminologie der Software als „Topobase“ bezeichnet. Es ist daher selbstverständlich möglich, eine Vielzahl von Usern, oder Topobases, anzulegen.

Die leere Grundstruktur einer Topobase besteht aus 51 Systemverwaltungstabellen, die im Anhang A aufgeführt sind. An dieser Stelle soll aber nur auf die wichtigsten Tabellen und Zusammenhänge eingegangen werden. Die Abbildungen der Tabellen wurden mit der Software Rose der Firma Rational erstellt.

Tabelle TB_Dictionary:

<<RelationalTable>> TB_DICTIONARY	
▶	F_TABLE_ID : NUMBER
▶	F_TABLE_CATALOG : VARCHAR2
▶	F_TABLE_SCHEMA : VARCHAR2
▶	F_TABLE_NAME : VARCHAR2
▶	F_GEOMETRY_COLUMN : VARCHAR2
▶	G_TABLE_CATALOG : VARCHAR2
▶	G_TABLE_SCHEMA : VARCHAR2
▶	G_TABLE_NAME : VARCHAR2
▶	STORAGE_TYPE : NUMBER
▶	GEOMETRY_TYPE : NUMBER
▶	COORD_DIMENSION : NUMBER
▶	MAX_PPR : NUMBER
▶	SRID : NUMBER
▶	TABLE_TYPE : NUMBER
▶	TOPIC_NAME : VARCHAR2
▶	F_TABLE_NAME_L1 : VARCHAR2
▶	TOPIC_NAME_L1 : VARCHAR2
▶	IDENT_SELECT_ATTRIBUTE_L1 : VARCHAR2
▶	TB_POINT_NUMBER : NUMBER
▶	HERKUNFT : NUMBER
▶	TABLE_DOES_NOT_EXIST : NUMBER
▶	ACTIVE : NUMBER
▶	SYS_C005409 = F_TABLE_ID
▶	SYS_C005411 = F_TABLE_NAME_L1
▶	SYS_C005413 = TOPIC_NAME_L1

Abbildung 3-6: Tabelle TB_Dictionary

In dieser Tabelle wird neben dem Aliasnamen für Tabellen auch der Tabellentyp aufgeführt. Der Tabellentyp kann folgende Werte annehmen:

Table_Type	Tabelle
1	Feature (Geometrie)
2	*_TBD Aufzählungstabelle (Domain Tabelle)
3	Attributtabelle (FID)
4	Kennungslisten (Kennung, Bezeichnung)
5	Attributtabelle (ID + weitere Felder)

Tabelle 3.5: Tabellentypen

Tabelle TB_Constraint:

<<RelationalTable>> TB_CONSTRAINT	
✓	CONSTRAINT_ID : NUMBER
✓	F_TABLE_NAME : VARCHAR2
✓	CONSTRAINT_STATEMENT : VARCHAR2
✓	ON_INSERT : NUMBER
✓	ON_UPDATE : NUMBER
✓	ON_DELETE : NUMBER
✓	ERROR_MESSAGE : VARCHAR2
✓	ERROR_CODE : NUMBER
✓	CONSTRAINT_NAME : VARCHAR2
✓	COLUMN_STRING : VARCHAR2
✓	ON_GEOMETRY_TABLE : NUMBER

Abbildung 3-7: Tabelle TB_Constraint

In dieser Tabelle werden Constraints (Regeln) auf Topobase-Tabellen verwaltet werden. Solche Regeln werden mit Hilfe des TB_Administrators definiert und werden bei Events wie Digitalisieren, Modifizieren oder Löschen wirksam. Für jede Regel wird in der Datenbank ein Trigger angelegt.

Tabelle TB_Relations:

<<RelationalTable>> TB_RELATIONS	
✓	ID : NUMBER
✓	PARENT_TABLE_NAME : VARCHAR2
✓	PARENT_COLUMN_NAME : VARCHAR2
✓	CHILD_TABLE_NAME : VARCHAR2
✓	CHILD_COLUMN_NAME : VARCHAR2
✓	DELETE_CHILD : NUMBER
✓	KEEP_RELATION : NUMBER
✓	ACTIVE : NUMBER
⚙	TB_RELATIONS_I1 = PARENT_TABLE_NAME
⚙	TB_RELATIONS_I2 = CHILD_TABLE_NAME
⚙	SYS_C005519 = ID

Abbildung 3-8: Tabelle TB_Relations

Diese Tabelle enthält Informationen über Relationen zwischen einzelnen Tabellen. Das Ausfüllen dieser Tabelle wird bei der Erstellung von einzelnen Featureklassen automatisch vom System durchgeführt.

Eine einfache Beziehung ist im folgenden Beispiel dargestellt:

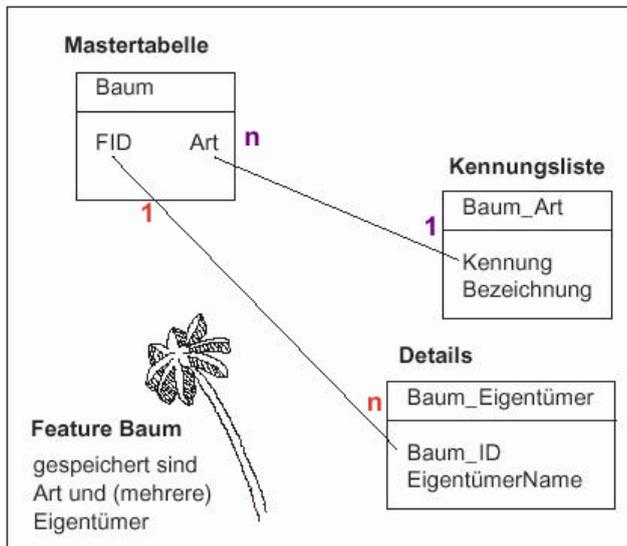


Abbildung 3-9: Beziehungen in der Topobase

In der Tabelle TB_Relations wird diese Beziehung folgendermaßen beschrieben.

	Spalte
Kennungslisten	
Baum	Child
Baum_Art	Parent
Master-Detail	
Link zu Detailtabelle	
Baum	Parent
Baum_Eigentümer	Child

In dieser Grundstruktur sind zunächst noch keine Geometrie- und Sachdaten enthalten. Dazu soll das Beispiel der „Linien“ wieder aufgegriffen werden, um zu

veranschaulichen, wie diese Daten in einer Topobase behandelt werden. Mit dem TB_Administrator wird eine neue Featureklasse „Linien“ angelegt. Dabei werden in der Oracle-Datenbank zwei Tabellen erzeugt.

Diese zwei Tabellen heißen

- Linien
- Linien_geo

In der Tabelle „Linien“ sind die Attribute gespeichert, während die Tabelle „Linien_geo“ die geometrischen Informationen enthält.

Dabei gehört zu jedem Feature eine GID (*Grafik-ID*), über die die Verknüpfung zur Grafik hergestellt wird.



Abbildung 3-10: Verknüpfung der Layer-Tabellen

Diese beiden Tabellen werden im weiteren Verlauf durch die Darstellung als Klassendiagramm und der Beschreibung der wichtigsten Felder etwas genauer erläutert. Die Datenablage der Geometrien entspricht im Groben dem relationalen Modell von Oracle.

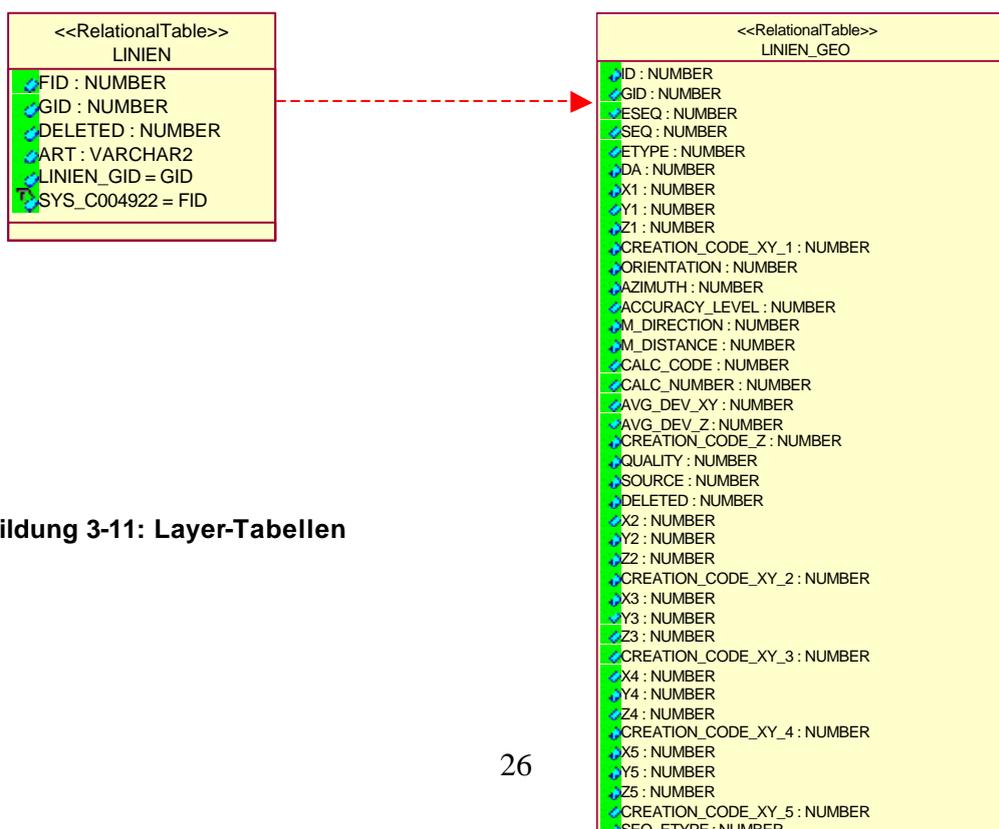


Abbildung 3-11: Layer-Tabellen

Die Featuretabelle „Linien“:

FID	Feature-ID: Jedes Objekt bekommt eine eindeutige ID
GID	Grafik-ID: Jedes Objekt bekommt eine GID. Über diese GID wird die *.geo-Tabelle angesprochen, in der wiederum mehrere Segmente des Objektes unter einer GID abgelegt sein können (<i>siehe relationales Modell von Oracle</i>)
Deleted	Ein gelöschttes Objekt wird zunächst nicht aus der Datenbank endgültig gelöscht, sondern erhält hier zunächst den Wert „1“. Somit wird es nicht weiter berücksichtigt, kann aber jederzeit wieder angesprochen werden.
Art	Ein Attributfeld für die Linienart
...	Weitere beliebige Attributfelder

Tabelle 3.6: Featuretabelle "Linien"

Die Geometrietabelle „Linien_geo“:

X1, Y1	Koordinaten
Z1	Höhe
Z_Terrain	Geländehöhe, die als Standardfeld neben der Höhe Z verwaltet wird.
Calc_Code	Entstehungscode
AVG_DEV_XY	Mittlerer Fehler (Lage)
AVG_DEV_Z	Mittlerer Fehler (Höhe)
Creation_Code_XY	Bestimmungsklassencode (Punkte)
Creation_Code_Z	Bestimmungsklassencode (Punkte)
Z_Diff	Überdeckungen/Höhendifferenzen
Accuracy_Level	Genauigkeit (Maßstab oder Genauigkeitsstufe)
Source	Herkunft
Deleted	1 = gelöscht; oder 0 = nicht gelöscht
TD_Point_Number	Punktnummer aus Migration Topodat
DA	Darstellungsart
Label_Text	Formatierter Text des Labels
Spatial_Index	Geometrie Index, der automatisch angelegt wird für eine Kachelungstiefe von 5

Tabelle 3.7: Geometrietabelle "Linien_geo"

Aus diesen Darstellungen wird ersichtlich, dass der Anwender bei der Verwendung der Software nicht zwangsweise mit der expliziten Erstellung komplizierter Datenmodelle konfrontiert wird. Auch sind zunächst keine umfangreichen SQL-Kenntnisse notwendig, da die komplette Erstellung und Erweiterung der Datenstruktur mit Hilfe des TB_Administrators vollzogen werden kann. Die einzigen Überlegungen die er durchdenken muss, sind die Bezeichnungen der einzelnen Featureklassen, sowie deren Ergänzung mit Aufzählungstabellen und Labelthemen (C-Plan (c), 2002).

Eine komplette Struktur einer Topobase bezogen auf die benutzerdefinierten Tabellen sieht beispielsweise folgendermaßen aus. Diese Ansicht ergibt sich aus einer Darstellung des TB_Administrators.

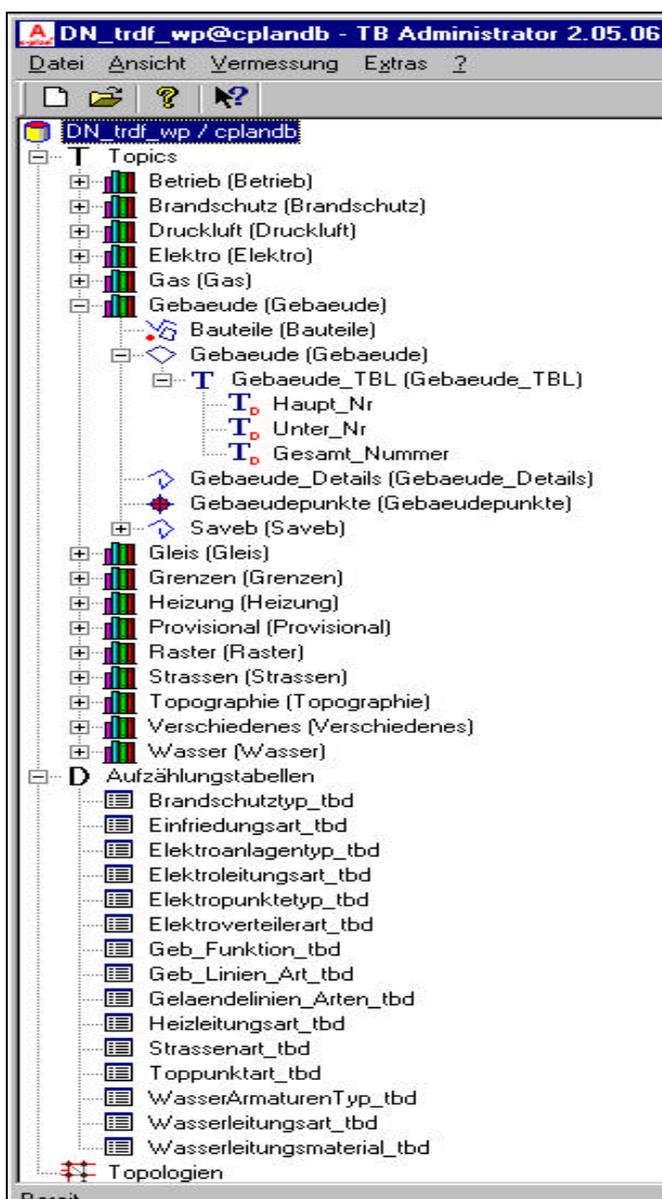


Abbildung 3-12: Komplette Struktur einer Topobase

3.3 Client

Als Client-Software kommen generell verschiedene Produkte in Betracht. Grundsätzlich ist dabei zwischen einer Betrachtung im Internet und einer Verwendung der Daten im Intranet bzw. einer Einzelplatzlösung zu unterscheiden. Zunächst ist die Entscheidung für einen Client zu treffen, der es erlaubt, die Daten von einem Einzelplatz aus zu pflegen und zu verwalten. Dieser Client sollte durch die Software AutoCad Map gebildet werden, da an diesem Arbeitsplatz nicht nur die Verwaltung der Daten, sondern auch die Erfassung und Fortführung des Datenbestandes im Vordergrund steht. Diese Software besitzt umfangreiche CAD-Funktionalitäten für die planerischen Tätigkeiten und Konstruktionen, bietet aber auch gleichzeitig durch den Aufsatz „Map“ GIS-typische Analyse- und Abfragemechanismen. Für die Internetlösung kommt die Software Autodesk MapGuide in Frage, da hier ein optimales Zusammenspiel mit AutoCad Map und auch den Middlewareprodukten gewährleistet ist.

Diese Software ist ein Paket aus mehreren einzelnen Applikationen, die in ihrem Zusammenwirken mit weiteren Komponenten ein Auskunftssystem bilden können. Die einzelnen Kernprodukte der Autodesk MapGuide-Suite werden nachfolgend beschrieben.

Autodesk MapGuide Author:

Mit Hilfe des Authors können sogenannte MWF-Dateien (MapWindowFile; Kartenfensterdatei) erzeugt werden. Diese Datei enthält vollständige Spezifikationen über den Inhalt und das Aussehen einer Karte, wie z.B. Kartenbegrenzung, Hintergrundfarben, Informationen zur Datenquelle, Layer mit kartographischen Daten, Anzeigeinformationen für die Layer, oder Menüs und Legenden. Anschließend können eine, oder mehrere MWF-Dateien in eine Webseite eingebettet werden.

MWF-Dateien werden standardmäßig im binären Format abgespeichert. Mit dem Author ist es auch möglich, die Informationen, die eine MWF-Datei enthält, im MWX-Format abzuspeichern. Dabei handelt es sich um ein Textformat, bei dem XML-Tags und -Knoten verwendet werden. Diese Dateien können mit dem Author, oder einem beliebigen XML-Parser geändert werden.

Autodesk MapGuide Viewer:

Durch den Viewer werden Karten im WWW-Browser angezeigt, wenn eine Website mit einer eingebetteten MWF-Datei, oder eine Verknüpfung mit einer solchen Datei aktiviert wird. Von diesem Programmteil gibt es drei verschiedene Versionen.

- Ein Plug-In für die Verwendung des NetscapeNavigators
- Ein ActiveX Control für die Verwendung des Microsoft Internet Explorers
- Eine Java-Edition für die Verwendung des NetscapeNavigators auf Sun-Solaris-Systemen, oder des Microsoft Internet Explorers auf Apple Macintosh-Systemen.

Autodesk MapGuide Server:

Der MapGuide Server verarbeitet Daten in Reaktion auf Anforderungen des Authors, oder der MapGuide Viewers, wenn eine Karte in diesen Programmen angezeigt werden soll. Der Server prüft die Anfrage und stellt die angeforderten Daten aus den möglichen Datenquellen zur Verfügung.

Der MapGuide Server umfasst die drei Komponenten MapAgent, ServerService und MapGuide Server Admin.

Das Zusammenwirken der Autodesk-Produkte im Einklang mit weiteren Komponenten ist in der folgenden Darstellung aufgezeigt (Autodesk, 2002).

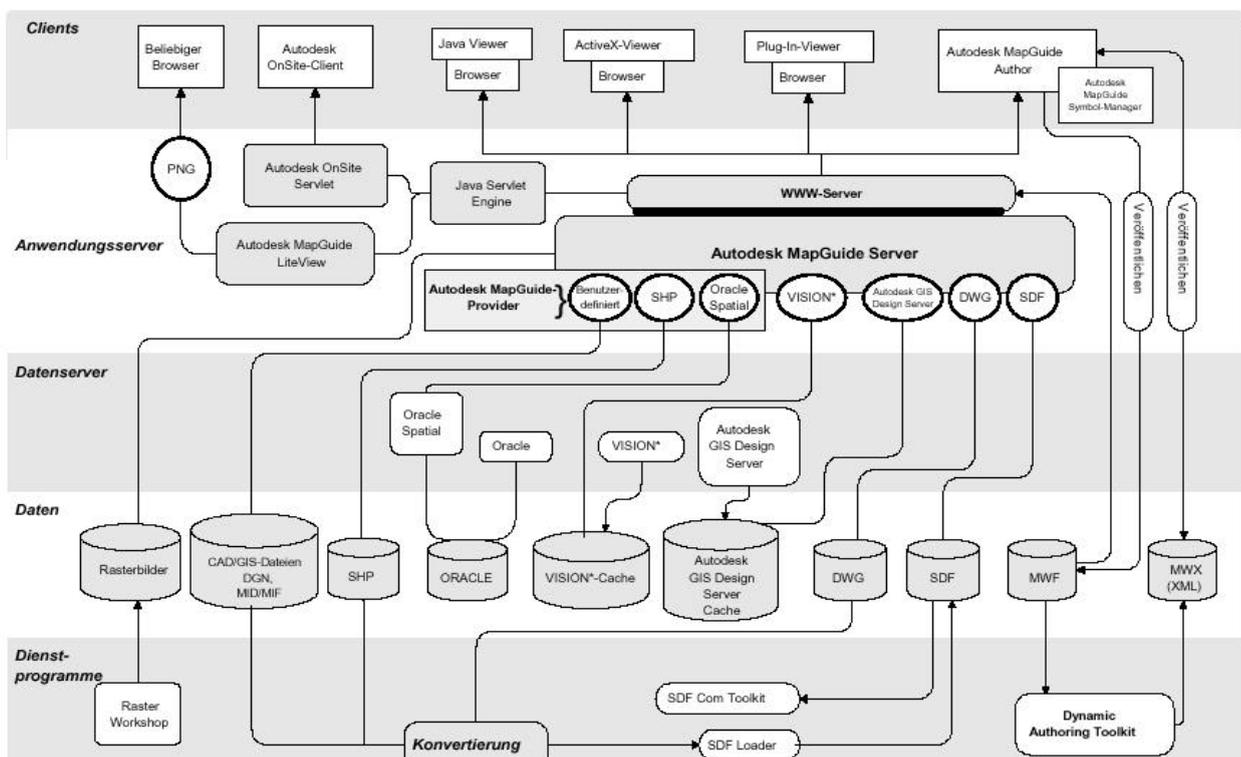


Abbildung 3-13: Zusammenwirken der Autodesk-Produkte

An dieser Stelle soll nicht auf alle Bestandteile des Schaubildes eingegangen werden, sondern nur das Prinzip des Datenflusses und die Zuständigkeiten der wichtigsten Komponenten erläutert werden.

Ein Client, wie z.B. ein Web-Browser, stellt eine Anforderung, nach der eine Karte angezeigt werden soll. Diese Anfrage wird von einem Web-Server aufgenommen. Diesen Web-Server stellt der Internet-InformationServer von Microsoft dar. Ein Web-Server ist eine Anwendung, die von einem Host-Computer ausgeführt wird. Dieser nimmt Anforderungen von anderen Computern auf, und verarbeitet und versendet verschiedene Arten von Dateien, wie z.B. Internetseiten, oder MWF-Dateien. Solche Anforderungen werden an einen Applikationsserver weitergeleitet, der im konkreten Fall durch den Autodesk MapGuide Server dargestellt wird. Werden andere Datenquellen angefragt, also keine MWF-Dateien, werden diese vom MapGuide Server bereitgestellt. Diese Datenquellen können z.B. dwg-Zeichnungen, SpatialDataFiles (SDF), Shapedateien, oder auch Daten aus einer Oracle (-Spatial) Datenbank sein. SDF-Dateien enthalten Daten, die aus anderen Datenquellen, wie Oracle, Shape, oder dwg erstellt werden können. In einer MWF-Datei hingegen sind keine Daten gespeichert, sondern nur u.a. die Information über den Speicherort der eigentlichen Daten. Eine MWF-Datei kann daher durchaus auf eine, oder mehrere SDF-Dateien als Datenquelle verweisen, was in der Darstellung nicht unbedingt deutlich wird.

Diese Darstellungen lassen schon vermuten, dass sich dieses Komponentengefüge sehr gut dazu eignen könnte, die migrierten Daten aus der Oracle Datenbank auf Auskunftsarbeitsplätze zu verteilen. Zum einen kann auf die Oracle Datenbank vom Applikationsserver direkt zugegriffen werden und zum anderen kann man sich SDF-Dateien erzeugen, wodurch nicht auf die originalen Daten direkt zugegriffen wird, sondern auf ein Abbild. Da es auch möglich ist, eine SDF-Datei direkt aus AutoCad Map von einer Zeichnung zu erstellen, könnte man immer wiederkehrende thematische Darstellungen eines Sachverhaltes auf diese Weise weitergeben.

3.4 Zusammenfassung

Die heutige „state of the art“-Aufteilung eines GI-Systems beruht auf der 3-tier-Architektur. Die drei Schichten bestehend aus Datenhaltung, Middleware und Client bilden die Grundlage für ein modernes und offenes GIS. Die Softwarelösung von C-Plan basiert auf Oracle 8i als Datenhaltungskomponente. Dieses DBMS bietet eine sehr gute Basis durch eine standardisierte Datenhaltung. Durch die zusätzliche Erweiterung „Spatial Cartridge“ lassen sich in Oracle räumliche Daten ablegen, indizieren, analysieren und abfragen. Mittels der räumlichen Indizierung werden Daten in den Tabellen wesentlich schneller ansprechbar. Die Indizierung kann entweder mit der Quadtree-, oder mit der R-Tree-Methode durchgeführt werden, wobei die Software der Firma C-Plan die erstere Methode verwendet. Generell werden in Oracle 8i noch das relationale Datenmodell und das objektrelationale Modell verwendet. Der objektrelationale Ansatz bietet einige Vorteile gegenüber dem relationalen Modell. Es können damit komplexere Geometrien gespeichert werden, wie zusammengesetzte Linienzüge oder zusammengesetzte Polygone. Zudem stellt dieser Ansatz eine Erleichterung bei der Erzeugung von Indizes und bei der Durchführung von räumlichen Abfragen.

Der grundlegende Unterschied ist, dass beim objektrelationalen Modell die Geometrie nur in einem einzigen Datenbankfeld abgelegt wird, während beim relationalen Ansatz jedes Element einer Geometrie in einer Datenzeile gespeichert wird.

Als Middleware wird das Produkt Topobase von C-Plan eingesetzt. Diese Entwicklung ist das Nachfolgeprodukt der Topodat, obwohl es softwaretechnisch betrachtet kaum mehr etwas mit der Topodat gemeinsam hat. Clientseitig kann auf die Topobase mit AutoCad Map, aber auch mit ArcView oder MapGuide als Internetlösung zugegriffen werden. Durch den modularen Aufbau können bei der Topobase verschiedene Fachdisziplinen bedient werden. Einzelne Fachschalen stellen hierfür vordefinierte Datenmodelle und Zugriffsmenüs zur Verfügung. Das Datenmodell der Topodat enthält eine Vielzahl von Metadaten Tabellen sowie pro Featureklasse zwei Tabellen. Die Geometrieinformationen einer Featureklasse werden in der Tabelle „layer_geo“, die Sachdaten in der Tabelle „layer“ abgelegt. Die Ergänzung des Metadatenmodells, welches vom System

komplett automatisch erstellt wird, um die benutzerdefinierten Featurotabellen wird dem Anwender durch das Modul TB_Administrator erleichtert.

Der Client wird in diesem Projekt durch AutoCad Map gebildet. Die Entscheidung für diese Software liegt in der Notwendigkeit, den Datenbestand ständig um neue Konstruktionen und Planungen zu ergänzen. Dafür ist es notwendig, entsprechende CAD-Funktionalitäten zur Verfügung zu haben. Solche umfassenden Möglichkeiten werden von anderen Clients kaum geboten. Für die Bereitstellung von internetbasierenden Auskunftsarbeitsplätzen wird als Client Autodesk MapGuide verwendet. Diese Software eignet sich sehr gut, da das Zusammenspiel mit dem Einzelplatz-Client unproblematisch ist und die Topobase ebenfalls ergänzende Module für diesen Client bereitstellt.

4 Ansätze zur Datenmigration

Überall dort, wo in einem Unternehmen mehr als ein (GIS-)System im Einsatz ist, stellt sich die Frage nach geeigneten Schnittstellen oder Austauschformaten, wenn diese Systeme ihre Daten untereinander austauschen sollen. Es gilt stets geeignete Methoden und Ansätze zu finden, um Daten von einem Quell- in ein Zielsystem zu überführen, ganz gleich, ob es dabei mehrere oder nur ein Quell- bzw. Zielsystem gibt. Zum einen darf bei einer Datenkonvertierung möglichst kein Informationsverlust stattfinden, so dass so wenig wie möglich im Zielsystem nachgearbeitet werden muss. Auf der anderen Seite möchte man aber auch bei der Konvertierung flexibel bleiben, um auf eventuelle Veränderungen des Quelldatenbestandes reagieren zu können. Grundsätzlich lassen sich zwei Möglichkeiten einer Datenkonvertierung ableiten: Einerseits über eine Direktschnittstelle und andererseits über ein Zwischenformat (Bärk, 2002).

4.1 Direktschnittstelle

Eine Direktschnittstelle kann als direkte Konvertierung der Daten des Quellsystems in die Daten des Zielsystems ausgelegt sein, oder aber auch in Form eines Online-Datenaustausches durch eine Datenservertechnologie. Das bedeutet, dass genau ein Quell- und genau ein Zielsystem betrachtet werden (1:1). Solche Schnittstellen sind in der Regel laufzeitoptimiert und bieten eine einfache Bedienbarkeit, da sie speziell auf die Bedürfnisse der beiden Systeme abgestimmt sind. Ist eine solche Schnittstelle erst konzipiert, kann die eigentliche Datenumsetzung schnell und unkompliziert ablaufen. Der große Nachteil dieser Schnittstellen ist aber, dass sie keineswegs flexibel sind. Ein veränderter Datenbestand wird sofort abgewiesen oder gar stillschweigend ignoriert. Weiterhin kommt hinzu, dass bei mehreren Quell- oder Zielsystemen die Anzahl der Konverter erheblich ansteigt. Bei vier verschiedenen Systemen, die alle untereinander kommunizieren sollen, sind schon 24 unidirektionale Konverter erforderlich.

4.2 Zwischenformat

Bei der Datenkonvertierung über ein Zwischenformat gibt es zwei Konzepte.

- Konvertierung der Quelldaten in ein systemneutrales Datenformat. Anschließende Konvertierung in das Zielsystem.

Bei dieser Möglichkeit wird aus beliebig vielen Systemen ein neutrales Austauschformat definiert (n:1). Ein neutrales Austauschformat als Bindeglied hat den Vorteil, dass bei mehreren Systemen, die Daten untereinander austauschen sollen, weniger Konverter benötigt werden. Bei vier Systemen werden so nur 8 Konverter notwendig sein. Diese Möglichkeit hat aber auch den Nachteil, dass stets eine komplette Kopie des Datenbestandes vorhanden ist, wenn die Austauschdateien nicht nach dem Übertragen wieder gelöscht werden. Ebenso ist es nachteilig, dass sich ein systemneutrales Austauschformat an den Möglichkeiten den schwächsten Systems richten muss, so dass es schon fast sicher ist, dass bei den „besseren“ Systemen Informationen bei der Konvertierung verloren gehen.

- Konvertierung der Quelldaten in ein einheitliches Datenformat und wahrer freier Zugriff aller Systeme auf dieses Datenformat.

Dieses durchaus erstrebenswerte Konzept lässt eine Konvertierung von beliebigen Quellsystemen in beliebige Zielsysteme über ein einheitliches Datenformat zu (n:m). Dadurch würden die zuvor gezeigten Nachteile aufgehoben, allerdings fordert diese Lösung ein einheitliches Produktmodell, wonach ein einheitliches Datenformat der Systeme vorhanden sein müsste.

Eine Möglichkeit des Datenaustausches über ein system- und anwenderneutrales Zwischenformat ist beispielsweise die Software CITRA der Firma CISS TDI GmbH. Dieses Format dient als Drehscheibe für verschiedenste Ein- und Ausgabeformate. In einem objektstrukturierten Aufbau der Schnittstelle werden die Objekte in einen Sachdaten- und Grafikbereich gegliedert, wobei der Teil der Informationen enthalten ist, der allen Systemen gemein ist. Es sind nur eine Sachdatenstruktur, aber mehrere grafische Primitive erlaubt. Die Identifikation

der Objekte erfolgt über neutrale Namen, durch die eine Verbindung zu den Einträgen in systemspezifischen Tabellen hergestellt werden kann. Durch eine Parametrisierung der Schnittstellen lassen sich diese vom Nutzer auf die eigenen Bedürfnisse anpassen. Durch das offengelegte Zwischenformat lassen sich Bestandsanalysen und Plausibilitätskontrollen durchführen. Es enthält lediglich ASCII-Zeichen, ist daher mit jedem Editor lesbar und bearbeitbar (CISS, 2002).

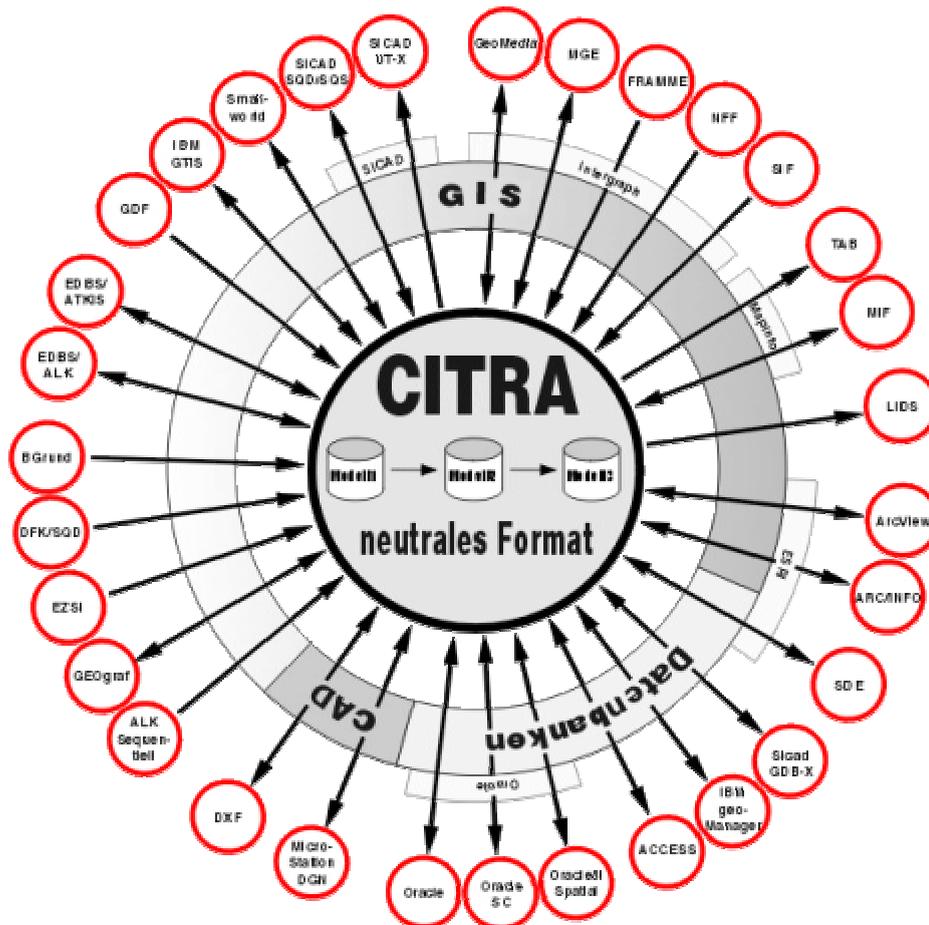


Abbildung 4-1: Austauschformat CITRA

Weiterhin sollte die Software FME der Firma SAFE erwähnt werden. Mit dieser Software ist es ebenfalls möglich, Datenbestände von einem System in ein anderes Format zu überführen. Auch FME definiert ein Zwischenformat, welches als Drehscheibe für eine Vielzahl an Formaten genutzt werden kann. Darüber

hinaus gibt es zahlreiche Möglichkeiten zur Analyse und Schnittstellendefinitionen (SAFE, 2002).

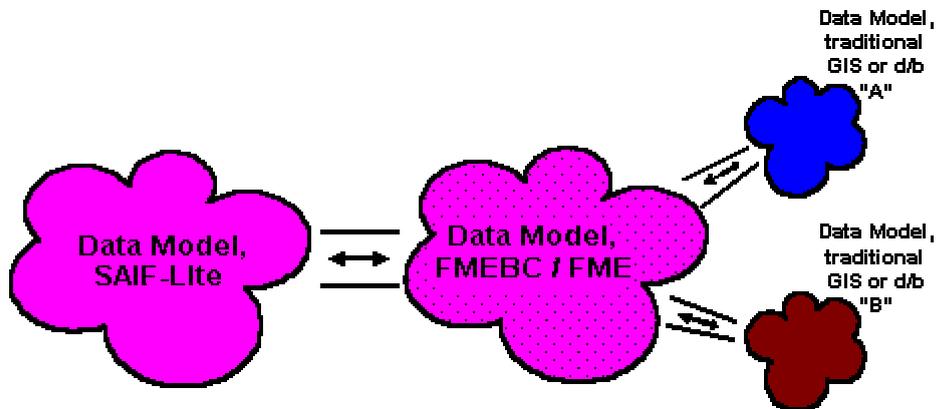


Abbildung 4-2: Datentransfer in FME

4.3 Bewertung

Eine Datenkonvertierung über ein Zwischenformat bietet gegenüber der Direktschnittstelle einige Vorteile. Bei der Umsetzung der Objekte in ein Zwischenformat ist es möglich eine Bestandsanalyse und Plausibilitätskontrollen durchzuführen. Durch eine Datenaufbereitung im Austauschformat wird eine hohe Datenqualität erreicht, die ein Nacharbeiten im Zielsystem nicht mehr, oder nur kaum erforderlich macht. Weiterhin kann ein neutrales Austauschformat als Drehscheibe zum Datenaustausch zwischen weiteren Systemen verwendet werden.

Für das konkrete Projekt der Datenmigration ist es ebenfalls sinnvoll, eine Konvertierung der Daten über ein Zwischenformat anzugehen. Vom Quellsystem werden die Daten zunächst in eine relationale Struktur in einer Oracle Datenbank abgelegt. Dort kann die Vollständigkeit und die Plausibilität der Daten sehr gut überprüft werden. Auch können an dieser Stelle eventuell notwendige Veränderungen am Datenbestand vorgenommen werden. Von diesem Zwischenformat aus werden die Daten in das Zielsystem überführt.

Eine Datenkonvertierung über eine Direktschnittstelle kommt hierbei keinesfalls in Frage. Wird einmal davon abgesehen, dass es für diesen Weg bislang keinerlei Schnittstellen gibt, diese also komplett neu entwickelt werden müssten,

sind die fehlenden Kontrollmöglichkeiten noch wesentlich ausschlaggebender. Der Aufwand, die Daten im Zielsystem nachzubearbeiten, muss noch zu dem Aufwand für die Definition der Schnittstelle gerechnet werden, wodurch ein nicht zu vertretender Gesamtaufwand entstehen würde.

5 Umsetzung des gewählten Ansatzes

Die Migration der Daten von der Topodat nach Topobase läuft in mehreren Schritten ab, die in diesem Kapitel beschrieben werden sollen. Zunächst wird ein grober Ablauf aufgezeigt, der in den weiteren Absätzen näher erläutert wird.

Die Durchführung einer solchen Migration setzt zwingend voraus, dass man sehr gute Kenntnisse über den Inhalt und den Aufbau des zu migrierenden Datenbestand besitzt. Auch ist es notwendig, dass man eine Vorstellung darüber gewinnt, wie die Datenablage in dem Zielsystem funktioniert. Erst dann ist gewährleistet, dass saubere und vor allem vollständige Datenübernahme durchgeführt wird.

5.1 Ablauf der Datenübernahme

Für die grundsätzlichen und entscheidenden Migrationsschritte bietet C-Plan vorgefertigte Tools an, mit deren Hilfe die Topodat-Daten in das Oracle-Umfeld überführt werden können. Das folgende Schaubild zeigt den schematischen Ablauf und die Einordnung der Werkzeuge, die bei der Migration Verwendung finden.

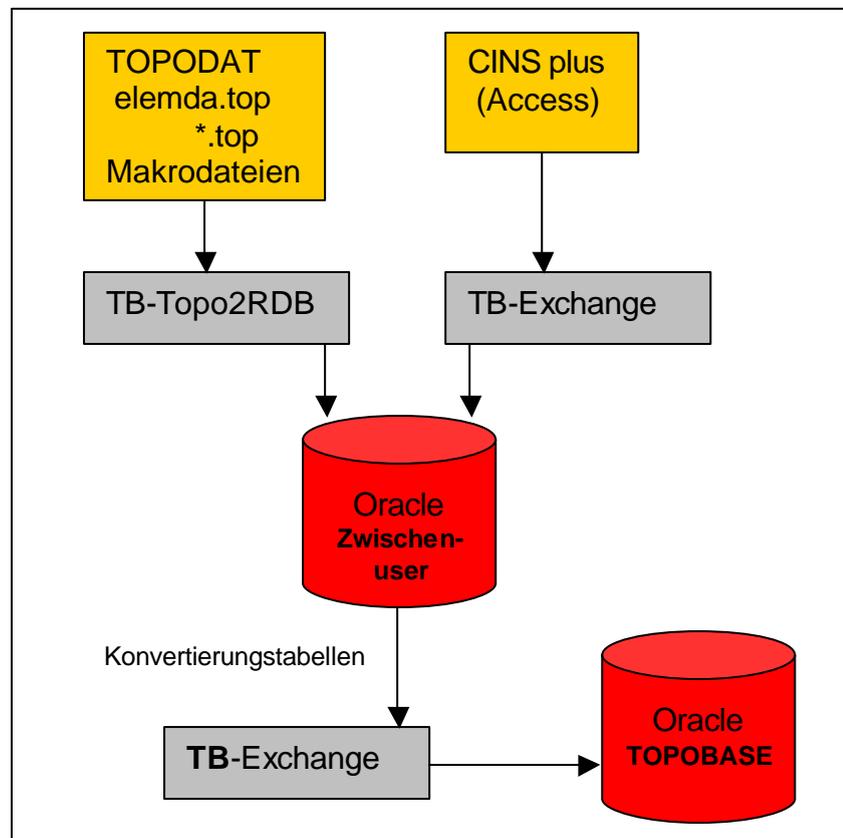


Abbildung 5-1: Ablauf der Datenübernahme

Bevor mit der Datenüberspielung begonnen wird, sollte man sich zunächst eine Statistik des Topodat-Datenbestandes ausgeben lassen. Führt man die dafür vorgesehene Funktion aus, erhält man eine Auflistung des Datenbestandes hinsichtlich verschiedener Sortierkriterien. Es wird beispielsweise ersichtlich, wie viele Punkte pro Vermarktungsart vorhanden sind, oder wie hoch die Anzahl der Linien einer bestimmten Linienart sind. Anhand einer solchen Statistik kann später sehr gut nachvollzogen werden, ob alle Daten mit übernommen worden sind. Auch dient diese Statistik später dazu, anhand der verschiedenen Vermarktungs- und Linienarten die neuen Featureklassen zu definieren, in die die jeweiligen Daten eingespielt werden.

Als erster Schritt wird damit begonnen, einen neuen User in Oracle anzulegen. Dieser User dient dazu, die Daten aus der Topodat in Oracle aufzunehmen. Im Endzustand werden diese Daten nicht mehr benötigt. Daher ist dieser User nur als Zwischenuser zu betrachten. Es empfiehlt sich, bei der Vergabe eines Namens, die Information mit einfließen zu lassen, dass es sich um einen Topodat-User handelt. Der alte Datenbestand war so strukturiert, dass es für jeden Standort des Konzerns ein eigenes Topodat-Projekt gab. Da sich diese Strategie in der Vergangenheit gut bewährt hat, soll auch im Oracle-Umfeld für jeden Standort ein eigener User angelegt werden. Weiterhin wurde festgelegt, dass die Daten, die sich auf das Abwassersystem beziehen, ebenfalls in einem eigenen User abgelegt werden, da dadurch eine übersichtlichere Handhabung der Daten aus Nutzersicht gegeben ist.

Mit Hilfe des Tools „TB-Topo2RDB“ (*Topobase to Relationale Datenbank*) werden sämtliche Daten aus den *.top-Dateien und den Makrodateien in eine einfache 1:1 Struktur in den Zwischenuser übernommen. Die Kanaldaten, welche in der Access-Datenbank vorliegen, werden mit dem Tool „TB-Exchange“ in den Zwischenuser überführt.

Der Zwischenuser enthält nun eine Anzahl von Tabellen, die alle Daten aus der Topodat enthalten. Welche Tabellen das im Einzelnen sind, wird im nächsten Abschnitt beschrieben. Mit Hilfe von Konvertierungstabellen, die zwar vom Programm TB-Exchange erstellt werden, aber vom Anwender befüllt werden müssen, werden im Anschluss die Daten vom Zwischenuser in den endgültigen User überführt. Dieser endgültige User besitzt dann die zuvor beschriebene Topobase-Struktur.

5.2 Datenübernahme in einen Zwischenuser

Um die Topodat-Daten in einen Oracle-Zwischenuser zu überspielen, werden zunächst alle Dateien der Topodat in ein Verzeichnis auf die Festplatte kopiert. Diese sind im einzelnen alle *.top-Dateien und die Makrodateien *.mac.

Für den Import unter deutscher Ländereinstellung muss im selben Datenverzeichnis eine Datei mit dem Namen Topo2RDB.inp angelegt werden. Diese Datei kann mit einem Texteditor erzeugt werden. Der Inhalt der Datei beschränkt sich lediglich auf die Textzeile „Komma = 1“ und sorgt dafür, dass die Koordinatenwerte mit Nachkommastellen korrekt eingelesen werden.



Abbildung 5-2: Datei Topo2RDB.inp

Sind diese Voraussetzungen erfüllt, kann das Programm Topo2RDB, welches sich im Topobase-Programmordner befindet, gestartet werden. Es öffnet sich folgendes Fenster, in dem mehrere Pfade angegeben werden müssen.

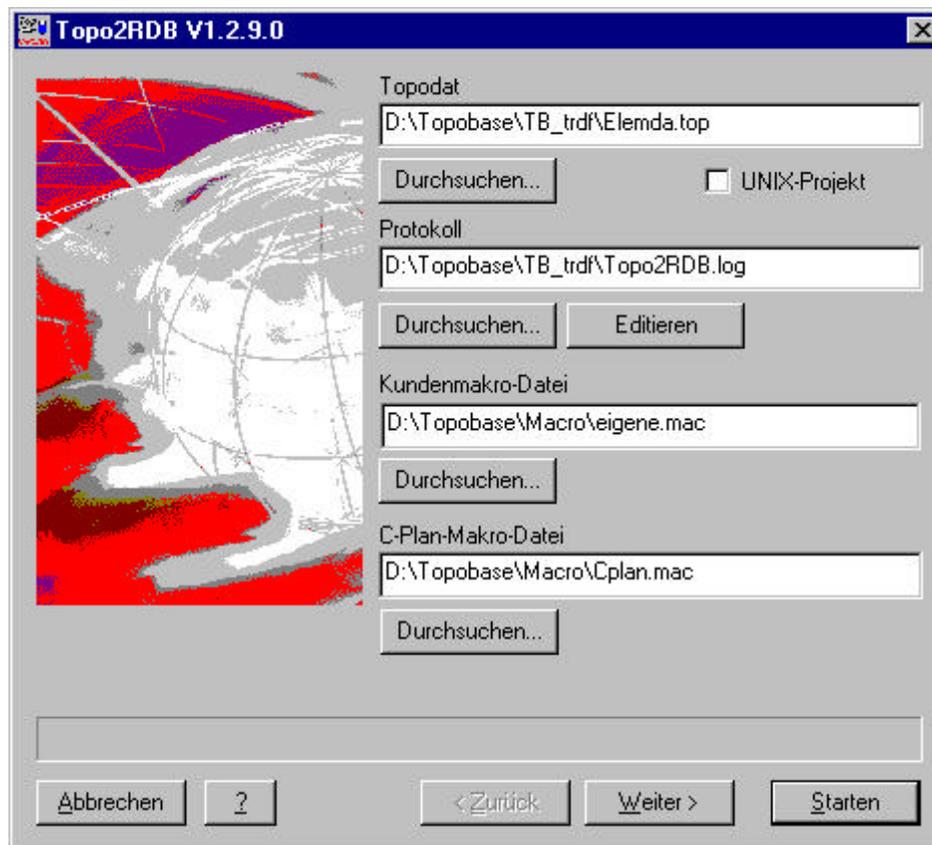


Abbildung 5-3: Programm Topo2RDB

Zuerst wird der Pfad zu den Topodat-Dateien benötigt. Hier genügt es, das Verzeichnis anzugeben. Der Ablauf des Imports wird in einer LOG-Datei dokumentiert, die in der nächsten Zeile definiert wird. Die unteren beiden Zeilen geben den Pfad zu den Makrodateien an, der in der Regel gleich dem oberen Pfad sein sollte.

Mit dem Fragezeichen-Button kann nochmals die Ländereinstellung überprüft werden. Die Einstellung muss hier in Übereinstimmung mit den Ländereinstellungen des Rechners sein.

Mit dem Weiter-Button öffnet sich das nächste Fenster, in dem ein Zeitfenster definiert werden kann. Da aber die gesamte Topodat übernommen werden soll, bleibt diese Option deaktiviert.

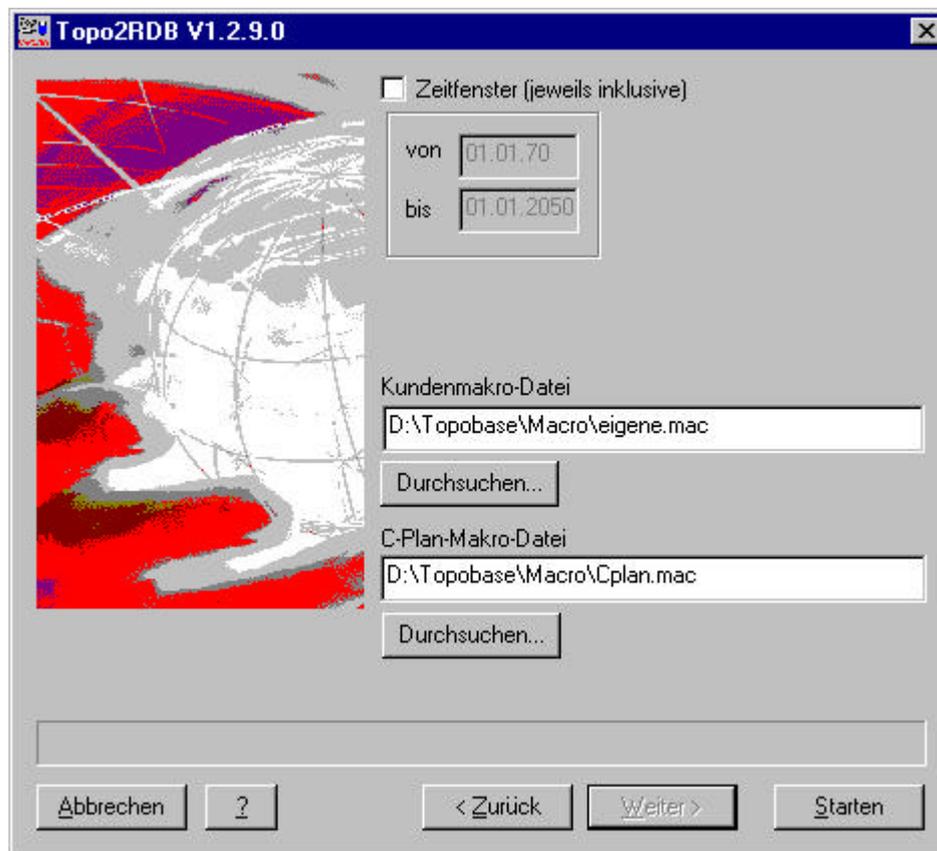


Abbildung 5-4: Zeitfenster in Topo2RDB

Mit dem drücken des Starten-Button wird die Datenbank für den Import vorbereitet. Es öffnet sich ein Oracle-Anmeldefenster für den temporären Zwischenuser, der als Vorbereitung bereits angelegt wurde. Sobald hier die Login-Daten angegeben wurden, besteht eine Verbindung zur Datenbank und der Import startet.

Über der Statusleiste werden die einzelnen Phasen der Übertragung angezeigt (Oracle DB öffnen, Import der Topodat, SQL Loader, ...).

Durch die Übertragung werden folgende Tabellen in dem Oracle-Zwischenuser angelegt:

TOPO_LINKAN ; TOPO_LINKOPF ; TOPO_MAKRO ; TOPO_MAKROGEO ;
TOPO_MASS ; TOPO_MASTER ; TOPO_POLYGON ; TOPO_POLYPKT ;
TOPO_PUNKT ; TOPO_TEXT

Da die Zwischentabellen nun in Oracle vorliegen, kann man sich diese entweder in Oracle selber, oder mit dem Tool TB_Generic anzeigen lassen. Ein Durcharbeiten aller Zwischentabellen ist durchaus sinnvoll, da man dadurch eventuelle Fehler, die beim Übertragen aufgetreten sind, beheben kann. Auch können an dieser Stelle in den Tabellen Ergänzungen oder Löschungen vorgenommen werden, wenn einzelnen Objekte nicht mit in den endgültigen Benutzer übernommen werden sollen.

Die wichtigsten Zwischentabellen sollen im folgenden Ablauf gezeigt werden.

Abbildung 5-5: Tabelle Topo_punkt

Die Zwischentabelle für Punkte zeigt die Anzahl der gespeicherten Punkte, die vorkommenden Vermarktungsarten, die Punktnummern, die Koordinaten, etc. Anhand dieser Tabelle und mit Hilfe verschiedener Filterfunktionen kann man überprüfen, ob die übertragenen Punkte übereinstimmen mit der Statistikauswertung, die man sich zu Beginn der Übernahme von der Topodat gemacht hat.

Topo_linkan				
Eid	Linieneid	Vorp.	Nachp.	Y1:
1	907	10340	10341	100!
2	907	10341	10342	100!
3	907	10342	10343	100!
4	907	10343	10344	100!
5	907	10344	10340	100!
*				

Abbildung 5-6: Tabelle Topo_linkopf

Die Zwischenspeicherung der Linien erfolgt in zwei Tabellen. Die Tabelle TOPO_LINKAN enthält die Kanten und die Tabelle TOPO_LINKKOPF enthält die Linienköpfe. Von der Tabelle TOPO_LINKKOPF besteht eine Relation über die Felder EID/LINIENEID zu den zugehörigen Linienkanten. Der im Beispiel gezeigte Linienzug ist geschlossen. Der erste Von-Punkt ist gleich dem letzten Nach-Punkt. Der Linienzug verläuft von Seqnr 1 bis Seqnr 5.

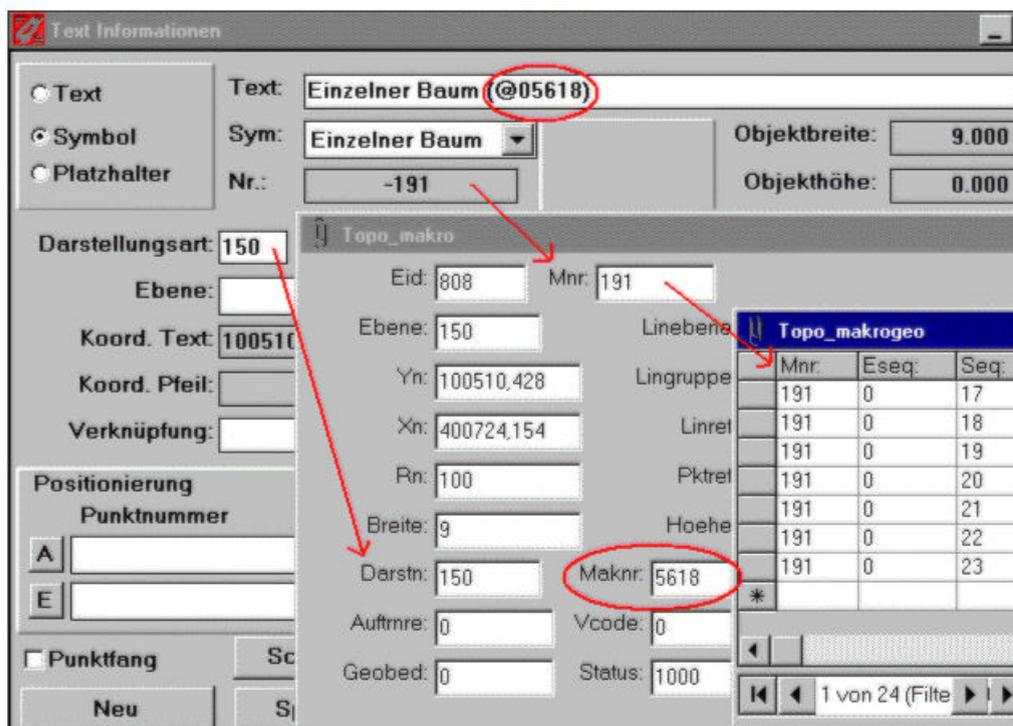


Abbildung 5-7: Tabelle Topo_makro

Die Makrodefinitionen werden ebenfalls in zwei Tabellen zwischengespeichert. Dabei besteht eine Relation zwischen den beiden Tabellen TOPO_MAKRO und TOPO_MAKROGEO. In der Tabelle TOPO_MAKRO wird die Definition mit der Makronummer und der Darstellungsart abgelegt, während die Tabelle TOPO_MAKROGEO die reine Geometrie des Symbols enthält.

Zusätzlich werden in dem Verzeichnis auf der Festplatte, in dem auch die Topodat-Dateien gespeichert sind, weitere Dateien hinzugefügt:

- ASCII-Dateien *.asc (Die Tabelleninhalten werden nochmals nach ASCII geschrieben, um diese auch außerhalb von Oracle kontrollieren zu können)

- Kontrolldateien *.ctr (Kontrolldateien für den Datenimport. Es sind die Felder aufgelistet, die mit den in der *.asc Datei durch Separatoren getrennten Werten gefüllt werden)
- Logdateien *.log (Der Importvorgang wird hier protokolliert)
- Fehlerdateien *.bad (Fehlerhafte Übertragungen werden hier aufgeführt)

Die Übertragung der Kanaldatenbank CINS plus kann in ähnlicher Weise erfolgen wie die vorherigen Daten. Auch dabei wird die komplette Access-Datenbank zunächst in einem Oracle-Zwischenuser abgebildet und dann in den endgültigen Benutzer übertragen. Der erste Schritt wird mit dem Programm TB_Exchange durchgeführt. Beim Starten des Programms zeigt sich zunächst folgende Oberfläche:

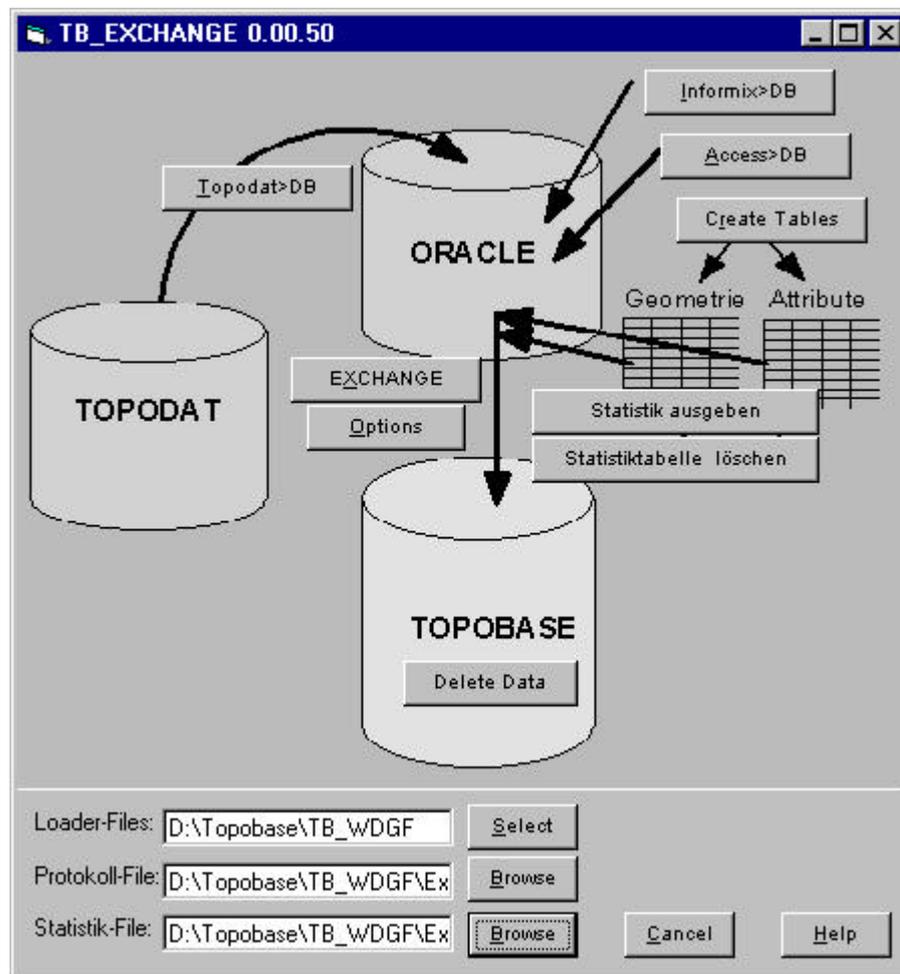


Abbildung 5-8: Programm TB_EXCHANGE

In diesem Fenster sind im unteren Teil Pfadangaben zu den Loader-Files zu machen. Diese Dateien werden vom Programm erstellt und werden benötigt,

um die Daten in den endgültigen Benutzer zu überführen. Weiterhin wird eine Protokoll- und eine Statistikdatei der Übertragung erstellt.

Startet man den Importvorgang mit dem Button „Access>DB“ wird man aufgefordert, die Access-Datenbank auszuwählen. Anschließend erfolgt die Login-Abfrage für den Zwischenuser der Oracle-Datenbank.

Auch bei dieser Übertragung werden wieder eine ganze Reihe von Tabellen sowohl im Oracle-Zwischenuser, als auch im angegebenen Verzeichnis auf der Festplatte erstellt, in denen die Access-Datenbank abgelegt wird. An dieser Stelle soll nicht weiter auf diese Tabellen eingegangen werden, da sich das Prinzip der Datenablage kaum von der zuvor beschriebenen Weise unterscheidet.

Speziell für die Kanaldaten soll aber noch darauf hingewiesen werden, dass es zudem eine weitere Möglichkeit gibt, eine Übertragung der Daten durchzuführen. Sowohl das Programm CINS plus als auch die Topobase-Umgebung haben eine Schnittstelle, über die Kanaldaten ausgetauscht werden können. Die sogenannte ISYBAU-Schnittstelle ist ein standardisiertes Austauschformat, welches aus einem Gemeinschaftsvorhaben des Bundes und der Länder resultiert, nach dem ein Datenaustausch zwischen verschiedenen Systemen gewährleistet werden soll.

Der Ablauf ist daher auf den ersten Blick weniger kompliziert, da man im Prinzip nur aus einem System exportieren muss, und anschließend in das Zielsystem importiert. Allerdings müssen bei dieser Variante die Daten im Ausgangs- und im Zielsystem gesichtet und kontrolliert werden, was eine gewisse Erfahrung mit Kanaldaten voraussetzt. Es gibt so gut wie keine Kontrollmöglichkeiten, ob der Datenaustausch fehlerfrei durchgeführt wurde. Bei der vorliegenden Projektaufgabe wurde für einen Industriestandort diese Variante verwendet. Im Endeffekt betrachtet wurde aber dadurch keine Aufwandsparnis erreicht.

5.3 Datenübernahme in den Enduser

Wenn die Zwischentabellen erstellt, kontrolliert und eventuell modifiziert sind, können die Daten in den Enduser übernommen werden.

An dieser Stelle beginnt die schwierigste und aufwendigste Arbeit der Migration. Es muss sich überlegt werden, wie der Datenbestand in der Topobase struktu-

riert werden soll. Während in der Topodat die Daten nach Ebenen, Gruppen und Nummern sortiert waren, werden die Daten in der Topobase nun eher objektorientiert abgelegt. Zwar konnte man sich in der Topodat einzelne Themengruppen definieren, denen dann einzelnen Datenbestände zugeordnet wurden, aber es handelte sich dabei nicht um eine themen- oder objektbezogene Speicherung der Daten an sich.

Daher ist es von essentieller Bedeutung, dass man den Datenbestand sehr gut kennt. Wie bereits beschrieben, ist es nicht notwendig ein umfangreiches Datenmodell zu entwickeln, aber das bereits von C-Plan vorgegebene Datenschema muss um die einzelnen Featureklassen, in denen die eigentlichen Daten gespeichert werden sollen, erweitert werden. Neben den einzelnen Themen, oder Featureklassen (*in Oracle: Layer*) muss sich auch überlegt werden, ob der alte Datenbestand hinsichtlich bestimmter Themen eventuell weiter untergliedert, oder auch verschiedentlich zusammengefasst werden kann.

Die Vorgehensweise der weiteren Übernahme ist

- Anlegen eines neuen Datenbankusers mit einer Topobase-Struktur
- Definition der Featuretabellen, Labeltabellen
- Einspielen der Daten in die einzelnen Featuretabellen

Diese einzelnen Schritte werden im weiteren Verlauf erläutert.

In Kapitel 3 wurde bereits beschrieben, dass man mit Hilfe des TB_Administrators einen neuen Datenbankuser anlegen kann. Dieser User besitzt dann eine Grundstruktur an Tabellen, die im Anschluss mit den Benutzer tabellen ergänzt wird.

Um mit dem TB_Administrator eine neue Topobase zu erstellen, darf nach dem Programmstart keine andere Topobase geöffnet sein. Im Kontextmenü der rechten Maustaste wählt man den Eintrag „Neue Topobase erstellen“. Es folgt ein Anmeldefenster für Oracle. Hier muss ein Oracle-User gewählt werden, der berechtigt ist, neue Benutzer anzulegen, z.B. „system/manager“. Im Anschluss öffnet sich folgendes Fenster, in dem die neue Topobase definiert wird.

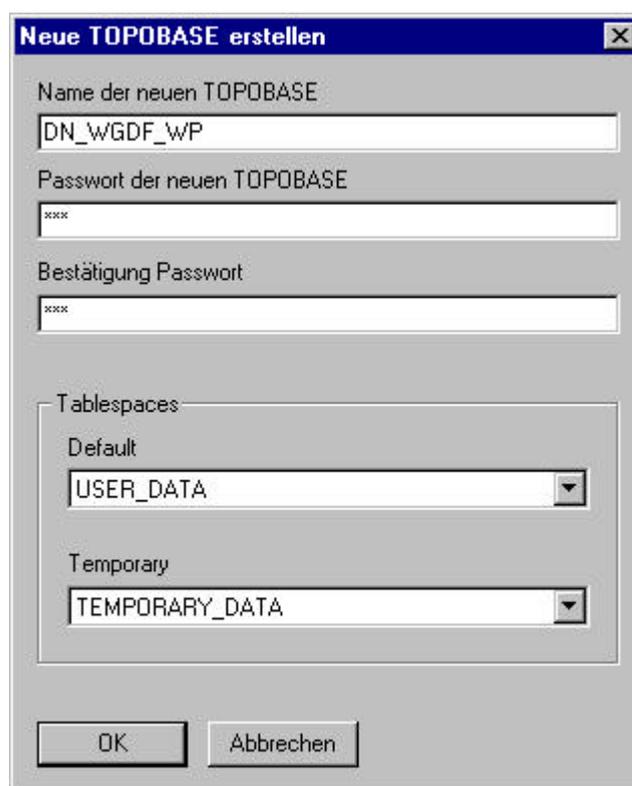


Abbildung 5-9: Neue Topobase erstellen

Hier werden Angaben zum Namen und Passwort der neuen Topobase verlangt, sowie Angaben zu den Tablespaces für Oracle. Bestätigt man dieses Fenster, wird die neue Topobase in Oracle angelegt.

Im Anschluss daran wird ebenfalls mit Hilfe des TB_Administrators die Tabellenstruktur für die Daten angelegt.

Um die Daten später in der Administrator-Ansicht und auch beim Zusammenstellen der einzelnen Featureklassen für einen Bildaufbau übersichtlicher zur Verfügung zu haben, werden sogenannte Topics erstellt. Diese dienen nur als Oberbegriffe für eine Anzahl an Featuretabellen.

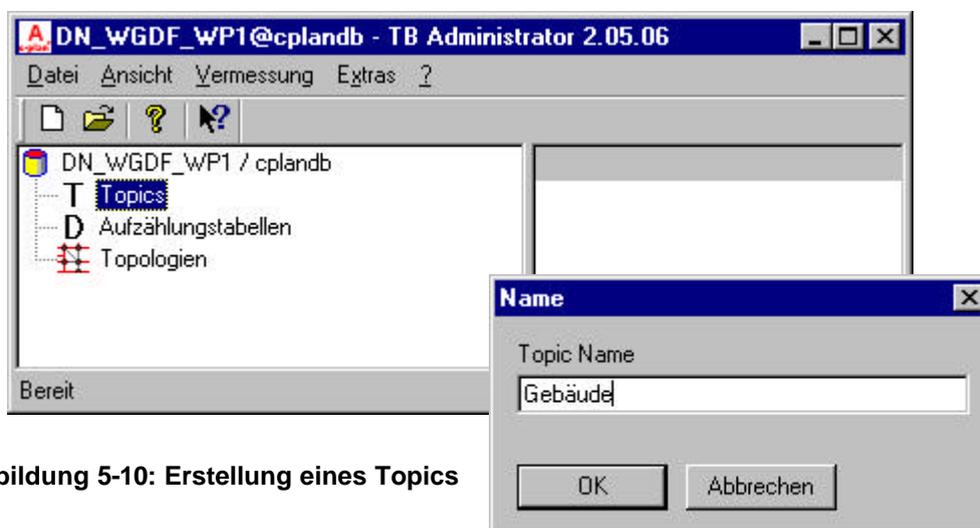


Abbildung 5-10: Erstellung eines Topics

Zu solchen Oberbegriffen werden nun die eigentlichen Tabellen erstellt.

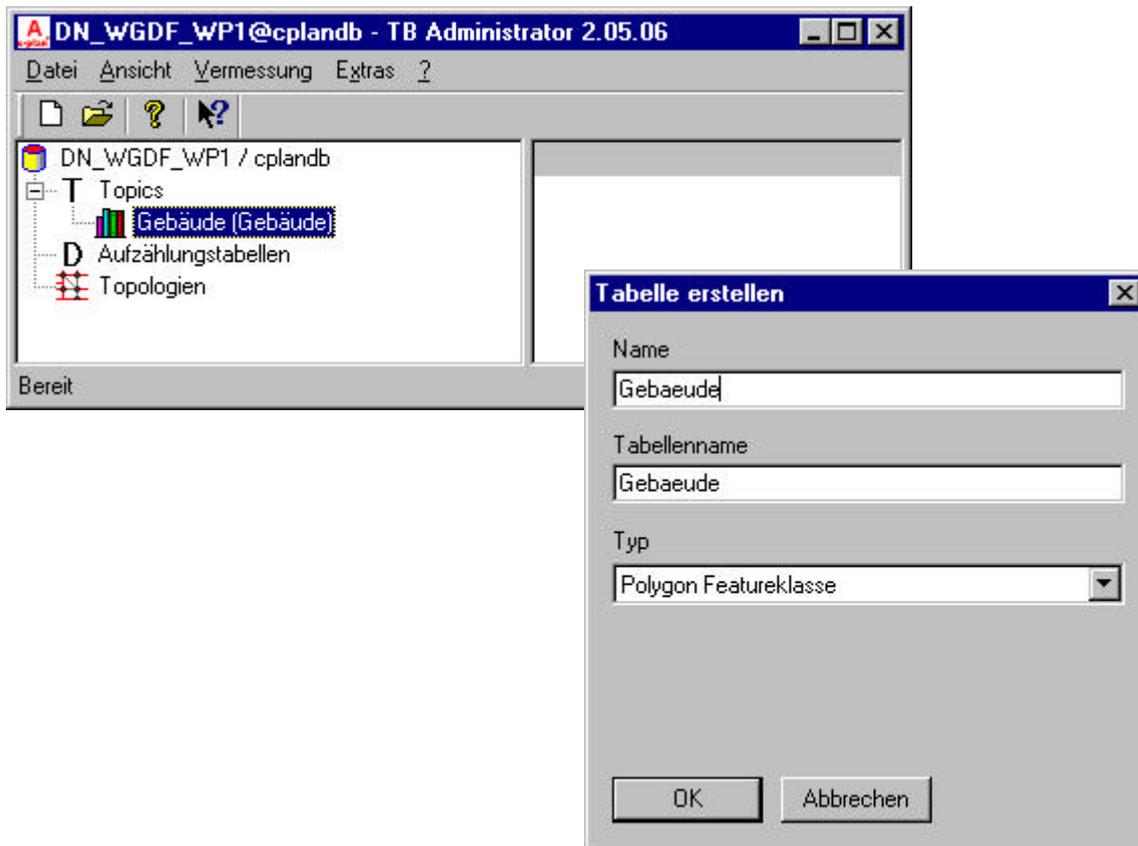


Abbildung 5-11: Erstellen einer Tabelle

Neben dem Tabellennamen muss hier auch der Datentyp angegeben werden, der einmal in der Tabelle abgelegt werden soll. Folgende Typen werden unterstützt:

- Punkte
- Polygone
- Linien(züge)
- Attributtabelle
- Area (Centroid inkl. Linestring)
- Collection (Punkte und Linien)
- Bemessungs Attributtabelle

Auf diese Weise kann man die gesamte Datenstruktur mit allen Tabellen, sowie eventuellen Relationen zueinander definieren. Die fertige Struktur, die sich auf die Gebäude bezieht, sieht z.B. folgendermaßen aus.

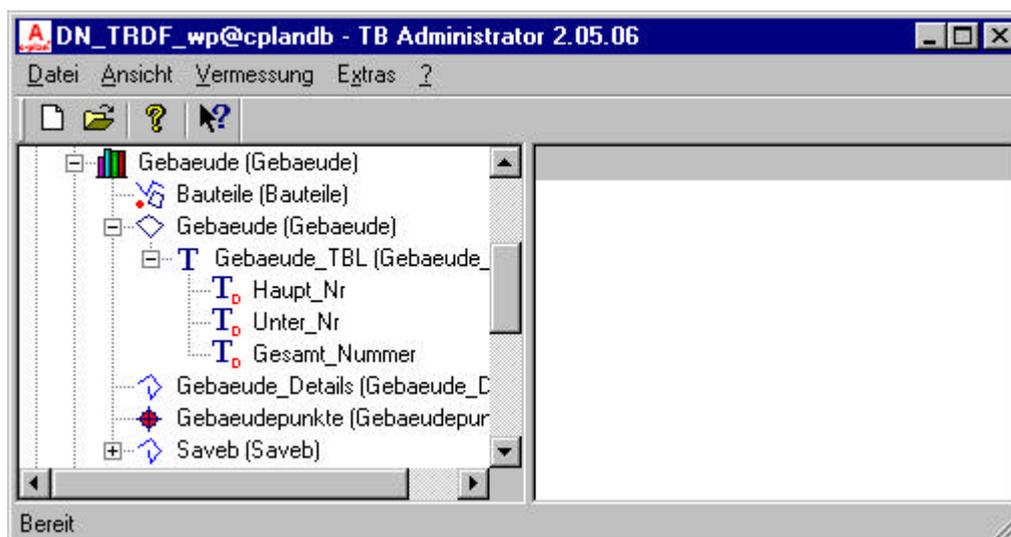


Abbildung 5-12: Struktur eines Topics

Für die Gebäudeanschriften wurden hier noch sogenannte Label-Featureklassen erstellt. Diese tbl-Tabellen enthalten die Definitionen für die Beschriftung der Gebäude.

Eine komplette Darstellung der erstellten Topics ist in Kapitel 3 enthalten. Die definierten Tabellen, sind im Anhang B aufgeführt.

Im weiteren Verlauf wird beschrieben, wie diese Tabellen prinzipiell befüllt werden. Dabei wird exemplarisch auf einige Tabellen eingegangen.

Generell hat es sich bewährt, nicht den ganzen Datenbestand auf einmal, sondern tabellenweise in den endgültigen User zu überführen. Besonders wenn der Datenbestand sehr groß ist, verliert man sehr schnell den Überblick, wenn man die Übernahmeparameter für den gesamten Bestand gleichzeitig definiert.

Wie in der Abbildung XX ersichtlich ist, wird für den letzten Schritt wieder das Tool TB_Exchange verwendet. Beim erstmaligen Befüllen einer Featuretabelle muss die Funktion „Create Tables“ im Hauptfenster des Programms durchgeführt werden. Diese Funktion erzeugt die sogenannten Konvertierungstabellen. In ihnen werden die Parameter der Übernahme festgelegt. Diese Konvertierungstabellen sind im einzelnen:

- Exchange_konv_table_geom
- Exchange_konv_table_attr
- Exchange_feature_konvert

In den ersten beiden Tabellen werden die Parameter für die Geometrien und die Attribute festgelegt, während in der letzten Tabelle die umgesetzten Daten eingetragen werden.

Im Folgenden werden die Tabellen `Exchange_konv_table_geom` und `Exchange_konv_table_attr` an einigen Beispielen erläutert.



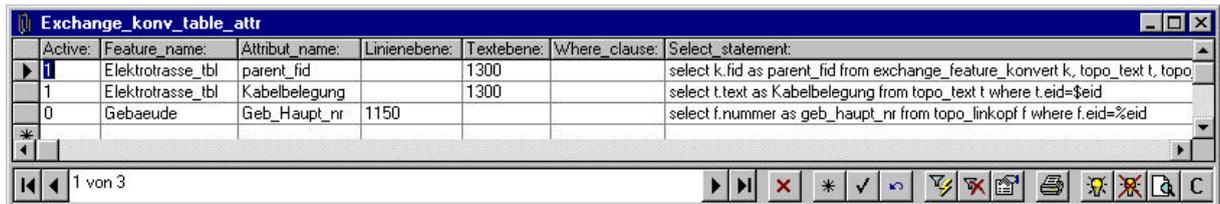
Activ	Exchange_seq	Feature_name	Label_name	Linienebene	Textebene	Where_clause	Select_statement	Bemerkung
0	1	Linien				ebene>0	select * from topo_linkopf	Alles
1	1	Elektrotrasse		1300				
1	2	Elektrotrasse_tbl	Kabelbelegung		1300			
0	1	Gebaeude		1150				Gebäude

Abbildung 5-13: Tabelle `Exchange_konv_table_geom`

In der Tabelle `Exchange_konv_table_geom` werden die Definitionen für die Übernahme der Geometrien eingetragen. Durch die erste Zeile werden alle Linien in eine Featureklasse namens „Linien“ übernommen. Obwohl die verschiedenen Linien noch in ihre verschiedenen Tabellen abgelegt werden, ist dies durchaus sinnvoll, da man am Ende der Migration dadurch erkennen kann, ob es noch Linien gibt, die noch nicht in spezifischen Tabellen abgelegt worden sind. Dies gilt für Punkte und Texte gleichermaßen. Die erste Spalte „Active“ lässt ein Ein- und Ausschalten der jeweiligen Zeile zu. Dadurch muss nicht die ausgefüllte Zeile gelöscht werden, wenn die Daten dieser Zeile bereits übernommen wurden. Durch die Definition einer „Exchange_seq“(uenz) kann die Reihenfolge der Bearbeitung gesetzt werden, wenn man mehrere Zeilen gleichzeitig umsetzen möchte. In der Spalte „Feature_name“ wird der Tabellename der Zieltabelle angegeben. Die Spalte „Linienebene“ (*Liniengruppe und Liniennummer*) lässt eine Ansprache der Objekte nach ihrer alten Einordnung „Ebene, Gruppe, Nummer“ zu. Durch die Felder „Where_clause“ und „Select_statement“ kann ein SQL-Statement angegeben werden, durch das sich der umzusetzende Aussahlsatz näher definieren, bzw. einschränken lässt. Im ersten Beispiel werden also alle Linien, deren Ebeneneinteilung > 0 ist aus der Zwischentabelle „TOPO_LINKOPF“ angesprochen. Es braucht hier nicht noch extra die Tabelle TOPO_LINKAN aufgeführt zu werden, da ja eine Verknüpfung zu dieser Tabelle besteht und somit die Linien komplett übernommen werden und nicht etwa nur die Anfangs- und Endpunkte der Linien. In der zweiten Zeile steht kein SQL-

Statement, aber eine Linienebene. Hierbei werden alle Linien der Ebene „1300“ in die Featuretabelle „Elektrotrasse“ übernommen.

Etwas umfangreicher gestalten sich meistens die Definitionen für die Attributwerte. Oftmals müssen dabei erst Verknüpfungen zwischen Tabellen definiert werden, wenn Attribute aus dem Feld der einen Tabelle in das Feld einer anderen Tabelle unter Berücksichtigung von Nebenbedingungen geschrieben werden sollen.



Active	Feature_name	Attribut_name	Linienebene	Textebene	Where_clause	Select_statement
1	Elektrotrasse_tbl	parent_fid		1300		select k.fid as parent_fid from exchange_feature_konvert k, topo_text t, topo
1	Elektrotrasse_tbl	Kabelbelegung		1300		select t.text as Kabelbelegung from topo_text t where t.eid=\$eid
0	Gebaeude	Geb_Haupt_nr	1150			select f.nummer as geb_haupt_nr from topo_linkopf f where f.eid=%eid

Abbildung 5-14: Tabelle Exchange_konv_table_attr

Mit der ersten Zeile wird eine Prozedur definiert, mit der das Attribut „parent_id“ der Labeltabelle „Elektrotrasse_tbl“ aufgefüllt wird. Das SQL-Statement dazu lautet:

```
select k.fid as parent_fid from exchange_feature_konvert k, topo_text t, topo_linkopf l where (l.ebene || l.gruppe || l.nummer)= (t.linebene || t.lingruppe || t.linnummer) and k.eid=l.eid
```

Die Angabe dieser Statements erfordert eine sehr hohe Sorgfalt, da eine falsche Eingabe durchaus dazu führen kann, dass Ursprungsdaten nicht in die dafür vorgesehenen Tabellen geschrieben werden, was im schlimmsten Fall nicht zwingend sofort auffallen muss.

Wenn eine oder mehrere Zeilen in den Konvertierungstabellen definiert wurden, kann der Importvorgang gestartet werden. Im Programmfenster von TB_Exchange wird der EXCHANGE-Button angeklickt, worauf das Programm die sogenannten Loader-Files erstellt. Das sind Dateien, die wieder die Endungen *.asc, *.log und *.ctl haben. Weiterhin wird eine Batch-Datei erstellt, mit deren Hilfe die Loader-Files nach Oracle geschrieben werden. Durch diesen Zwischenschritt ist es möglich, anhand der Loader-Files zu überprüfen, ob die richtigen Daten an die richtige Stelle geschrieben werden. Ansonsten wäre kein

Überprüfen der Angaben in den Konvertierungstabellen möglich. Der Inhalt der Batch-Datei sieht z.B. folgendermaßen aus:

```
sqlldr USERID=dn_wgdf/***@cplandb,  
CONTROL=D:\Topobase\TB_WDGF\Loader\AW_HALTUNG_feature.ctf,  
LOG=D:\Topobase\TB_WDGF\Loader\AW_HALTUNG_feature.log,  
DATA=D:\Topobase\TB_WDGF\Loader\AW_HALTUNG.asc
```

Diese kann nach Überprüfung gestartet werden, wodurch die Daten letztendlich in den endgültigen Benutzer nach Oracle geschrieben werden.

5.4 Zusammenfassung

Der gewählte Ansatz, die Daten über ein Zwischenformat zu migrieren, hat sich als sehr vorteilhaft herausgestellt. Durch das Anlegen eines Zwischenusers in der Oracle Datenbank lassen sich gute Kontrollmöglichkeiten nutzen, um die Daten verlustfrei zu übernehmen. Durch einfache Vergleichsabfragen ist es möglich, eventuelle Differenzen zu einer zuvor angefertigten statistischen Auswertung des Quelldatenbestandes auffindig zu machen und zu beheben. Das Werkzeug Topo2RDB, mit dem die Quelldaten in einer einfachen 1:1 Struktur im Zwischenuser abgelegt werden, hat sich als sehr gut erwiesen, da Unstimmigkeiten bei der Übertragung nicht ignoriert, sondern in einer Fehlerdatei festgehalten werden.

Für den Aufbau der Datenstruktur des endgültigen Topobase Users hat sich das Programm TB_Administrator als sehr hilfreich herausgestellt. Damit ist es mit wenig Aufwand möglich, eine Tabellenstruktur in der Oracle Datenbank aufzubauen. Auch umfangreichere Tabellenverknüpfungen können durch das Definieren von Aufzählungs- und Labeltabellen durchgeführt werden. Mit Hilfe von Konvertierungstabellen werden die Daten vom Zwischenuser in den endgültigen User überführt. Durch das Definieren der Übertragungsregeln mit der standardisierten Abfragesprache SQL wird eine sehr flexible Möglichkeit gegeben, auch implizierte Datenstrukturen zu übertragen. Durch diesen Standard hat es sich in der Praxis bewährt, auch andere Werkzeuge wie Access für Teildefinitionen und -Übernahmen zu verwenden.

6 Einsatz der erzeugten Daten

Der umgesetzte Datenbestand soll neben seiner Nutzung als Planungs- und Entscheidungsgrundlage mit dem AutoCad Map – Client auch weiteren Verwendungen zugeführt werden. Dabei geht es um die Bereitstellung von Auskunftssystemen auf der Basis von MapGuide, was als Hauptziel galt. Dieser originäre Nutzen sollte umgehend nach der Datenmigration realisiert werden. Weiterhin geht es um den Einsatz der Daten in Drittsystemen und somit der Prüfung in Richtung Standardisierung.

6.1 Originärer Nutzen der Daten

In diesem Kapitel soll beschrieben werden, wie und in welcher Form die Oracle-Daten einem Client zur Verfügung gestellt werden können. Durchaus dazu gehörende Fragestellungen bezüglich Sicherheitsaspekte, Auswahl der Client/Server-Verbindung (Internet, ISDN-Router, etc.), erweiterte Programmierung von Zugangsberechtigungen und Benutzeranpassungen werden nicht umfassend, oder gar nicht behandelt, da dies den Rahmen dieser Arbeit übersteigen würde. Es soll die generelle Machbarkeit einer Implementierung aufgezeigt werden.

Dabei wird der Weg verfolgt, auf der Basis von SDF-Dateien die Daten dem Applikationsserver zugänglich zu machen. Es wird eine MWF-Datei erzeugt, die diese SDF-Dateien als Datenquelle verwendet. Anschließend wird mit dem Programm TB_GenericWeb eine Oberfläche zur Abfrage des Datenbestandes hergestellt. Dieser Ablauf wird im folgenden Schema dargestellt.

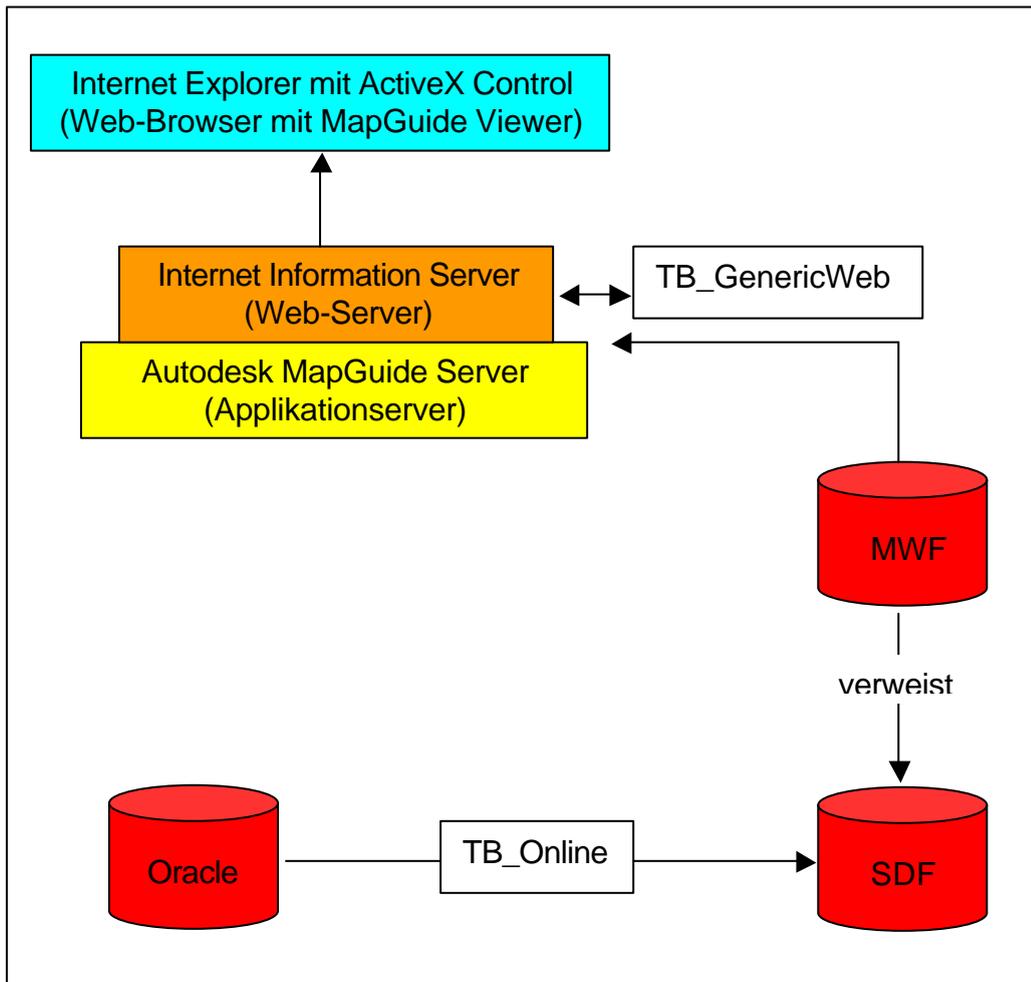


Abbildung 6-1: Von der Datenquelle zum Client

Zunächst werden mit Hilfe des Programms TB_Online die SDF-Dateien erzeugt. Dabei muss für jede Featureklasse (Layer) eine eigene SDF-Datei erstellt werden, die in ein separates Verzeichnis auf dem Server abgelegt werden. Mitunter kann es sehr unterschiedlich sein, in welchen zeitlichen Abständen eine Datenaktualisierung erfolgen muss. Wenn sich Daten stündlich, oder in noch kürzeren Abständen ändern, ist es sehr aufwendig, in gleichen Abständen die SDF-Dateien zu erzeugen. Daher bietet das Tool die Möglichkeit, in vorgegebenen Zeitrahmen diese Dateien automatisch zu erzeugen. Hier wird allerdings von dieser Funktion kein Gebrauch gemacht, da sich eine zeitliche Änderung des Datenbestandes lediglich im Wochenbereich bewegt. Man muss dabei auch bedenken, dass beim Verwenden der Funktion das Programm TB_Online stets gestartet sein muss, was alleine schon zu einer weiteren Belastung der Serverkapazität führt.

Mit dem MapGuide Author wird das Design der Karte und der Legende, sowie die Definitionen der Datenquellen erstellt. Diese Angaben werden anschließend in einer MWF-Datei abgespeichert. An dieser Stelle kann man sich bereits das Ergebnis der Kartenerstellung anschauen, indem die MWF-Datei in das Rootverzeichnis des Internet-Informationsservers kopiert, und anschließend mit dem Browser aufgerufen wird. Voraussetzung dafür ist, dass das ActiveX Control für den Internet Explorer, also der MapGuide Viewer, installiert ist.

Auf der folgenden Seite ist ein Screenshot einer testweise erstellten MWF-Datei dargestellt. Außer dem Hinzu- und Wegschalten von Legendeneinträgen sowie einigen Mapping-Funktionen, wie Pan, Vergrößern, oder Verkleinern sind keine weiteren Abfragen hier möglich.

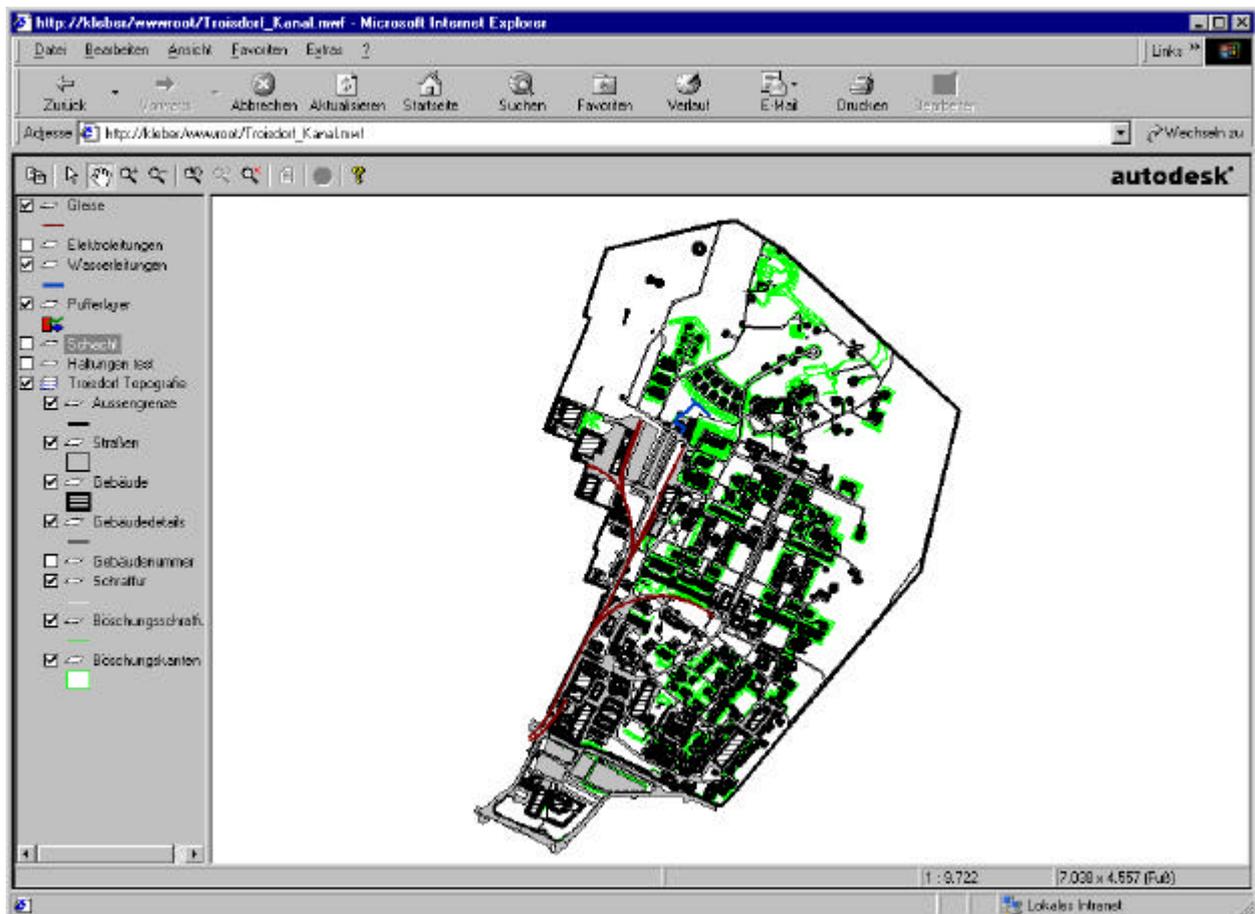


Abbildung 6-2: Ansicht einer mwf-Datei

Damit nun über den Client auch Abfragen durchgeführt, und Benutzerrechte vergeben werden können, wird das Programm TB_GenericWeb verwendet.

Im oberen Frame ist die Menüstruktur für einen Benutzer zu erkennen, der in der Hauptsache Kanaldaten zur Ansicht und Abfrage zur Verfügung gestellt bekommen hat. Wählt man einen Menüeintrag aus, werden im linken Frame die Unterpunkte angezeigt. Ein Klick auf den Schriftzug „Haltung“ öffnet beispielsweise die Tabelle für die Abwasserhaltungen in einem separaten Fenster. Da diese Frames alles ASP-Seiten sind, dazu noch in einer sehr einfachen und leicht zu durchschauenden Struktur vorliegen, kann die gesamte Seite nach belieben angepasst werden.

6.2 Externer Nutzen der Daten

Ein weiterhin erklärtes Ziel der Datenmigration ist es, eine standardisierte Datenthaltung zu gewinnen, die unabhängig von einem Systemhersteller ist. Es sollte zum einen möglich sein, den Datenbestand durch verschiedene Systeme zu nutzen. Eine weitere Fragestellung ist, ob man das System als interoperabel bezeichnen kann. Eine Definition von Interoperabilität ist z.B.:

„Interoperabilität bezeichnet die Möglichkeit, verschiedenartige Daten in einem einzelnen Arbeitsablauf zu integrieren. Dies setzt voraus, dass Syntax und Semantik der Daten dem Anwender in einheitlicher Form zur Verfügung gestellt werden. Interoperabilität erlaubt den transparenten Zugang zu mehreren raumbezogenen Daten- und Verarbeitungsrecourcen innerhalb eines einzigen Arbeitsablaufes, ohne sie in einen Datenbestand zu überführen.“ (Bill, 2001)

Daher stellt sich die Frage, ob sich der jetzige Datenbestand dazu eignet, von verschiedenen (GI-)Systemen gelesen, bearbeitet und analysiert werden kann und darüber hinaus auch gleichzeitig zu verwenden ist. Dieser Fragestellung soll nun nachgegangen werden, indem versucht wird, exemplarisch mit dem GIS GeoMedia Professional von INTERGRAPH die von C-Plan abgelegten Daten anzusprechen.

Da nun die Datenstruktur der Topobase hinreichend genau erläutert wurde, muss daher im Folgenden auf die Datenstruktur von GeoMedia Professional in Bezug auf Oracle Spatial eingegangen werden. Anschließend wird geprüft, ob es starke Parallelen zwischen diesen beiden Strukturen gibt, oder ob es große Unterschiede nicht zulassen, die Daten mit einem Fremdsystem zu fassen.

Die Software GeoMedia Professional kann Daten aus einer Oracle Spatial Datenbank lesen und auch in eine solche schreiben. Dabei ist es möglich, sowohl das relationale, als auch das objektrelationale Datenmodell zu verwenden. Bei schreibendem Zugriff auf eine Oracle Spatial Datenbank sind auf jeden Fall für beide Objektmodelle geomeadia-spezifische Metadaten erforderlich. Für den lesenden Zugriff sind diese nicht zwingend erforderlich, jedoch ratsam. Diese Metadaten werden bei beiden Objektmodellen automatisch beim Schreibzugriff erzeugt. Bei lesendem Zugriff werden diese Metadaten nicht automatisch erzeugt. GeoMedia Professional bietet aber hier die Möglichkeit, mit einem beigefügten Dienstprogramm die notwendigen Metadaten automatisch zu erzeugen. Dies wird allerdings ausschließlich für das objektrelationale Datenbankmodell angeboten. Da hierbei der Aufwand erheblich größer ist, wird dafür das sogenannte „GDOO-Package“ verwendet. Dieser Zusatz enthält eine Reihe von PL/SQL-Statements, die die Erstellung der Metadaten erleichtern.

6.2.1 Die Objektmodelle von GeoMedia Professional

Um die beiden Oracle Spatial Objektmodelle von GeoMedia Professional zu erläutern, soll wieder exemplarisch eine Ebene mit dem Namen und dem Typ „Linie“ abgelegt werden. Die beiden Objektmodelle sind Bestandteile des Befehlssatzes „Geographic Data Objects for Oracle (GDOO)“.

Zunächst wird auf das relationale Modell eingegangen, da die Software Topobase wie beschrieben ebenfalls relationale Strukturen zur räumlichen Datenablage verwendet.

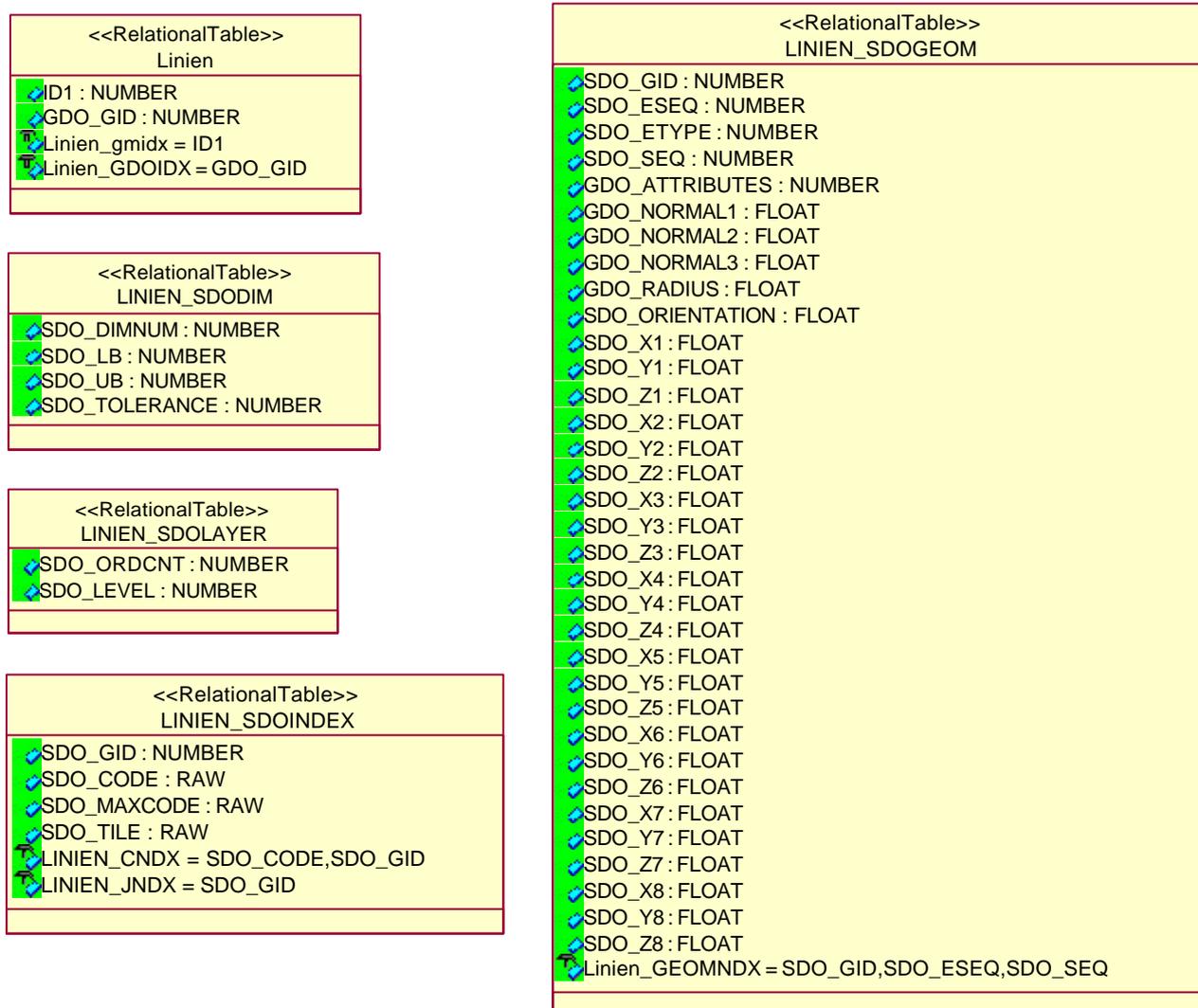


Abbildung 6-4: Relationale Modell von GeoMedia

Man sieht sehr deutlich, dass diese Struktur identisch ist mit der, die Oracle selbst für das relationale Modell vorgibt. Es sind hierbei sowohl die Tabellennamen, die Spaltennamen sowie deren Feldtyp wiederzufinden.

In GeoMedia Professional wird die Ebene „Linie“ als Objektklasse bezeichnet, während diese Objektklasse in Oracle „Layer“ genannt wird. Dieser Layer setzt sich aus zwei wichtigen Tabellen zusammen: „LINIEN“ und „LINIEN_SDOGEOM“.

In der Tabelle „Linien“ werden die Attributdaten für das Objekt gespeichert.

Die Tabelle „LINIEN_SDOGEOM“ enthält die Geometrieinformationen des Objektes. Diese können punkt-, linien- oder flächenförmig sein. Beim Verbindungsaufbau zu einer Oracle Datenbank sucht GeoMedia Professional diese beiden Tabellen, um sie als Objektklasse darzustellen.

Die drei weiteren dargestellten Tabellen sind erforderlich, um die räumliche Komponente zu berücksichtigen. Diese Tabellen spielen allerdings eher auf der Datenbankebene eine Rolle und werden von GeoMedia Professional nicht berücksichtigt.

Die Tabelle „LINIEN_SDOINDEX“ enthält die räumliche Indizierung des Layers. In der Tabelle „LINIEN_SDOLAYER“ sind die Informationen über die Kachelungstiefe des Layers enthalten. Die Informationen über die Begrenzung der Daten eines Layers sind in der Tabelle „LINIEN_SDODIM“ enthalten. Der minimale und maximale Wert ist wichtig für die Indizierung und die Unterteilung eines Layers.

Damit die Tabellen „LINIEN_SDOGEOM“, „LINIEN_SDOINDEX“ und die Attributtabelle zusammengeführt werden können, benötigt das Datenmodell von Oracle Spatial eine numerische Spalte mit dem Namen „SDO_GID“. Im GDOO-Modell heißt diese Spalte „GDO_GID“ und ist in der Attributtabelle zu finden.

Betrachtet man nun die Datenstruktur des objektrelationalen Modells, so stellt man auch hier fest, dass sich GeoMedia Professional an den von Oracle Spatial vorgegebenen Strukturen orientiert.

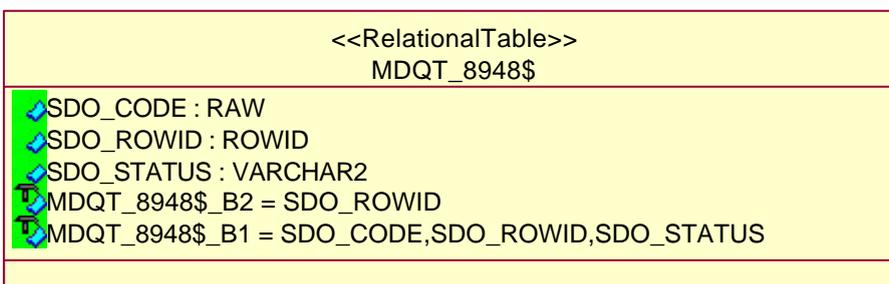
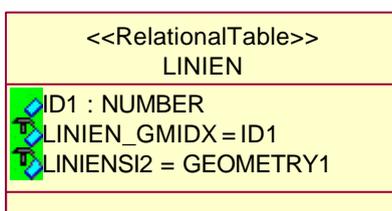


Abbildung 6-5: Objektrelationale Modell von GeoMedia

Hier gibt es zunächst einmal nur eine Tabelle, die den Namen der Objektklasse „LINIEN“ trägt. In dieser Tabelle gibt es ein Feld („GEOMETRY1“), in dem die Geometrie des Objektes abgelegt wurde. Dieses Feld hat den Feldtyp

„MDSYS.SDO_GEOMETRY“. Näheres zu diesem Feldtyp wurde bereits im Kapitel zu Oracle Spatial beschrieben.

Die zweite Tabelle „MDQT_8948\$“ beinhaltet die Information über den räumlichen Index, der nach der Quadtree-Methode ermittelt wurde.

6.2.2 Metadatenstruktur von GeoMedia Professional

Neben den standardmäßigen Vorgaben der Tabellen- und der Datenstrukturen sind für GeoMedia Professional noch weitere Daten erforderlich. Diese sogenannten Metadaten beschreiben die eigentlichen Daten und besitzen auch eine fest vorgegebene Struktur. Diese Strukturen unterscheiden sich beim relationalen und objektrelationalen Ansatz voneinander, so dass hier wieder beide Modelle getrennt voneinander betrachtet werden müssen.

Beim relationalen Ansatz werden die Metadaten im gleichen Datenbankschema abgelegt, wie die eigentlichen Daten selbst. Es werden eine Reihe von Tabellen erzeugt, die im weiteren Verlauf näher beschrieben werden sollen.

Die folgenden Metadatentabellen werden bei der Verbindung zu einer räumlich relationalen Datenbank erstellt:

- *ColumnMetadata*: In dieser Oracle-spezifischen Tabelle werden Informationen über die einzelnen räumlichen Layer, deren Geometrietyp und deren Projektion abgelegt.
- *GCoordSystem*: Diese Tabelle speichert die verwendeten Koordinatensystemdefinitionen
- *GAliasTable*: Durch einen Alias wandelt diese Tabelle spezifische Tabellennamen in die von GeoMedia Professional erwarteten Namen um.
- *ModificationLog*: In dieser Tabelle werden auf Wunsch alle Veränderungen eingetragen. Die Veränderungen werden mit Hilfe von Triggern aufgezeichnet.
- *ModifiedTables*: Speichert die veränderten Tabellen ab.

Weiterhin werden folgende Sequenzen erstellt:

- *GEOMEDIA_SEQ*: Diese Hauptsequenznummer wird verwendet, wenn ein eindeutiger Schlüssel benötigt wird, z.B. beim Festlegen von Indexspalten
- *GDDO_MODLOGSEQUENCE*: Generiert die Sequenzen für die Tabelle ModificationLog.
- *GEOMEDIAQW_SEQ*: Generiert die Sequenzen für die Abfragefenstertabellen

Weitere Metadatentabellen, die bei Bedarf erzeugt werden, sind:

- *GFeatures*: In dieser Tabelle werden alle Objektklassen aufgeführt, die mit GeoMedia Professional definiert worden sind. Wenn diese Tabelle vorhanden ist, dann sind in GeoMedia Professional auch nur die Objektklassen sichtbar, die in dieser Tabelle stehen.
- *AttributeProperties*: Diese Tabelle enthält die Darstellungseigenschaften aller nichtgrafischen Datenbankspalten.
- *GeometryProperties*: Diese Tabelle enthält die Darstellungseigenschaften für alle grafischen Objekte.
- *FieldLookup*: In dieser Tabelle ist die Methode zum Indizieren der Tabellen AttributeProperties und GeometryProperties gespeichert.

Auf der folgenden Seite sind alle Metatabellen des relationalen Modells dargestellt.

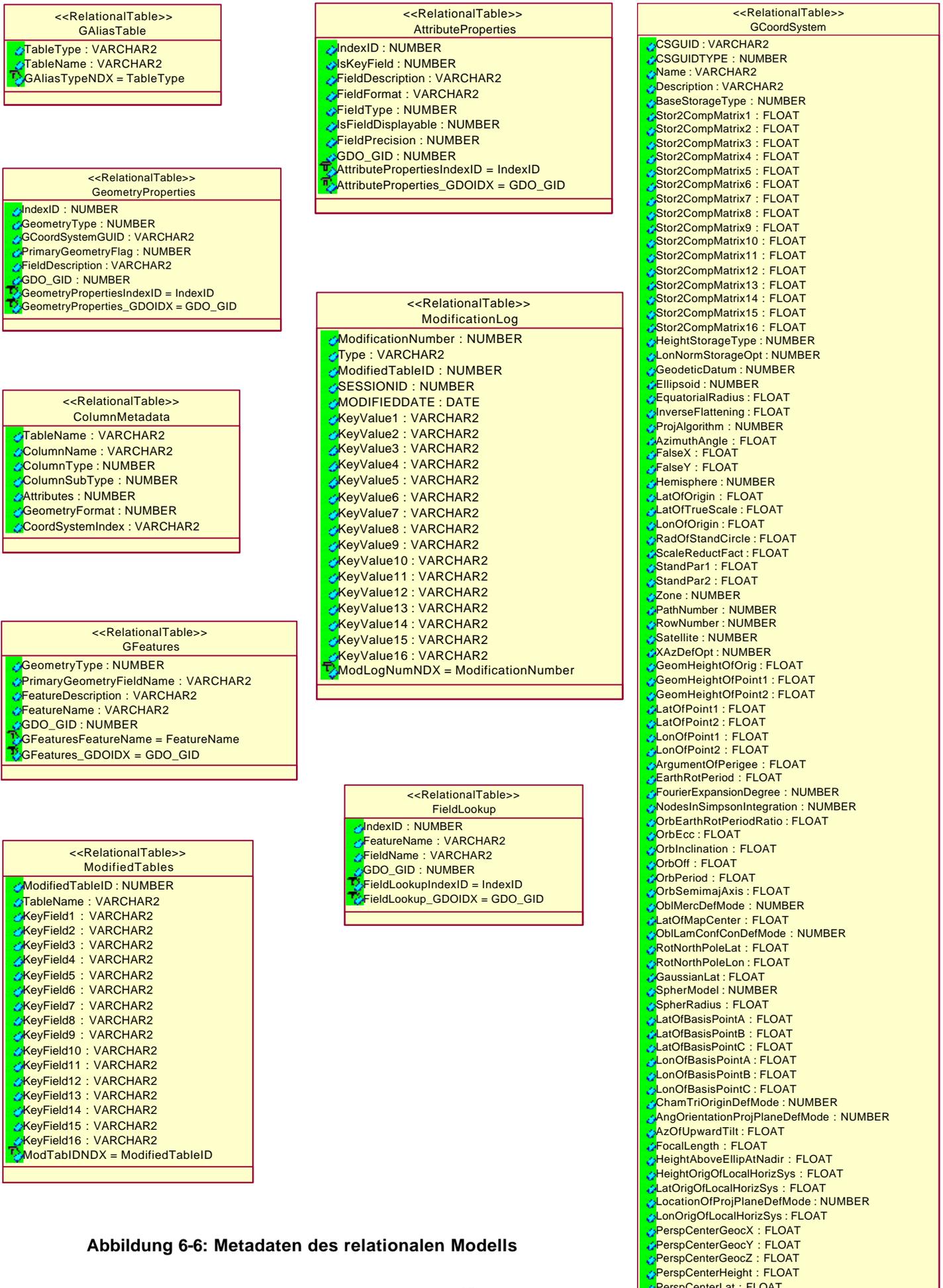


Abbildung 6-6: Metadaten des relationalen Modells

Ganz anders hingegen sieht die Metadatenstruktur beim objektrelationalen Modell aus. Hier werden die Metadaten nicht im gleichen Schema gespeichert wie die Daten selbst, sondern in einem eigens dafür definierten Schema. Dieses Schema heißt „GDOSYS“. Um dieses Schema zu erzeugen, gibt es generell zwei Möglichkeiten. Zum einen kann das SQL-Script „METADATEN.SQL“ benutzt werden. Dieses Script wird auch ausgeführt, wenn Daten im Programm GeoMedia Professional in das Oracle Objektmodell exportiert werden. Zum anderen kann das Datenbankdienstprogramm verwendet werden, welches zum Installationsumfang von GeoMedia Professional gehört.

Die folgenden Tabellen sind Bestandteile des Metadatenschemas GDOSYS.

- *ATTRIBUTEPROPERTIES*: In dieser Tabelle werden die Schlüsselinformationen, die Präzision und die Formate der Felder der Tabelle FIELDLOOKUP festgehalten.
- *FIELDLOOKUP*: Diese Tabelle stellt für jede Eigenschaft in jeder Objektklasse einen eindeutigen Bezeichner für das GeoMedia-Metadatenystem zur Verfügung. Zudem werden die Eigenschaftsnamen für alle Objektklassen im Format BENUTZERNAME.TABELLENNAME gespeichert, sowie die Spaltennamen für das zugeordnete Objekt.
- *GALIASTABLE*: Hier werden Namen von weiteren Standardtabellen und von den GeoMedia Client-Tabellen festgelegt. Weiterhin ist in dieser Tabelle festgelegt, welche Metadaten für einen Benutzer erforderlich sind. Die GALIASTABLE ist im Prinzip nur eine parametrisierte Ansicht einer anderen Tabelle, die eine Ansichtsdefinition für jeden Benutzer enthält. Somit kann jedem Benutzer durch eine andere Ansicht der Tabelle ein anderes Metadatenystem zur Verfügung gestellt werden.
- *GCOORDSYSTEM*: In dieser Tabelle werden die Koordinatensysteminformationen gespeichert. Wenn diese Tabelle nicht vorhanden ist, kann keine Koordinatentransformation durchgeführt werden. Stattdessen wird das Ansichtskordinatensystem von GeoMedia Professional benutzt. Jede Objektklasse kann in GeoMedia Professional ein eigenes Koordinatensystem aufweisen.

- **GEOMETRYPROPERTIES:** Diese Tabelle legt die GUID für das Koordinatensystem ab, welches von den Geometriefeldern eines Objektes verwendet wird, sowie weitere Eigenschaften dieser Geometriefelder. Weiterhin wird das Koordinatensystem abgespeichert, welches den einzelnen Objekten zuzuweisen ist.
- **GFEATURE:** In dieser Metatabelle werden alle Tabellennamen der Objektklassen gespeichert. Es können nur die Objektklassen in GeoMedia Professional sichtbar gemacht werden, die auch in dieser Tabelle aufgeführt sind. Auch diese Tabelle stellt nur eine parametrisierte Ansicht der Tabelle GFEATURESBASE dar, durch die der Benutzer nur solche Objekte sieht, auf die er Zugriff hat. Diese Ansicht der Tabelle (*ein sogenannter View*) wird mit dem folgenden SQL-Statement erzeugt:

```
CREATE VIEW GDOSYS.GFEATURES AS
SELECT * FROM GDOSYS.GFEATURESBASE WHERE FEATURENAME
IN
(SELECT OWNER || '.' || OBJECT_NAME FROM ALL_OBJECTS);
```

Dadurch kann man die Ansicht der Tabelle GFEATURESBASE beeinflussen, solange die gleiche Definition wie bei der Tabelle GFEATURES benutzt wird.

- **GFIELDMAPPING:** Diese Tabelle ist Oracle-spezifisch und wird nicht direkt von GeoMedia Professional benutzt. Hier werden verschiedene Aspekte der Felddefinitionen abgelegt, sowie der GDO-Datentyp, die Koordinatensystem-ID und alle verwendeten AutoNumber-Sequenzen der Spalten jeder Objektklasse.
- **GPARAMETERS:** Diese Parametertabelle wird nicht von GeoMedia Professional, sondern von Oracle bei der Erstellung eines Geometriefeldes benutzt. Anhand der Parameter in dieser Tabelle können beispielsweise Geometriefelder indiziert werden. Wenn diese Tabelle nicht vorhanden ist, oder kein Eintrag für diese Tabelle in der Tabelle GALIASTABLE vorhanden ist, kann kein Geometriefeld erzeugt werden. Abhängig davon, ob der Parameter „Z“ vorhanden ist, oder nicht, werden 2D-, oder 3D-Geometrien erzeugt.

- *MODIFIEDTABLES*: Die Funktion dieser Tabelle ist identisch mit der namensgleichen Tabelle im relationalen Metadatenmodell. Wenn diese Tabelle bei aktivierter Protokollierung nicht vorhanden ist, schlägt die Methode OpenDatabase fehl.
- *MODIFICATIONLOG*: Auch diese Tabelle entspricht in ihrer Funktion der entsprechenden Tabelle des relationalen Metadatenmodells. Da in dieser Tabelle alle Änderungen an einer Objektklasse festgehalten werden, wächst der Inhalt stetig an. Daher sollte diese Tabelle in regelmäßigen Abständen mit Hilfe des Datenbank-Dienstprogrammes, oder mit dem SQL-Statement

```
SQL> TRUNCATE TABLE GDOSYS.MODIFICATIONLOG
```

geleert werden. Auch hier kann die Änderungsprotokollierung über Trigger, oder über eine automatische Protokollierung des Datenservers erfolgen.

Des Weiteren werden verschiedene Sequenzen erstellt. Durch die Sequenzen werden AutoNumber-Felder ausgefüllt, die vom GDOSYS-Schema benutzt werden.

- *GMODLOG*: Diese Sequenz ist zuständig für das Feld MODIFICATIONNUMER in der Tabelle MODIFICATIONLOG.
- *GAUTONUMBERSEQUENZ*: Durch diese Sequenz wird ein eindeutiger Bezeichner zur Erzeugung weiterer Sequenzen erstellt.
- *FIELDLOOKUPINDEXID1*: Diese Sequenz ist zuständig für das Feld INDEXID in der Tabelle FIELDLOCKUP.

Standardmäßig wird ein Trigger erstellt, der die entsprechenden Metadaten im Schema GDOSYS aktualisiert, wenn eine Tabelle oder Spalte in der Oracle Datenbank gelöscht wird. Dieser Trigger heißt *DELETEMETADATAGMT*.

Die Metatabellenstruktur des Schemas GDOSYS ist im Anhang C dargestellt. Die Metatabellen stehen untereinander in Beziehung. Die direkten und die sich implizierenden Beziehungen sind dargestellt.

6.2.3 Zugriff auf den Datenbestand

In diesem Kapitel wird beschrieben, inwiefern die Software GeoMedia Professional in der Lage ist, auf den vorhandenen Datenbestand der Topobase zuzugreifen.

Vergleicht man die zuvor erläuterten Datenmodelle, stellt man fest, dass es zwischen der relationalen Datenstruktur der Topobase und der Datenstruktur von GeoMedia Professional durchaus Parallelen gibt. Der Vergleich im einzelnen sieht folgendermaßen aus:

	Topobase	GeoMedia Pro. (Rel. Modell)
Tabellennamen	Linien	Linien
	Linien_geo	LINIEN_SDOGEOM
Spaltennamen	GID, ESEQ, ...	SDO_GID, SDO_ESEQ, ...
Spaltentypen	Number	Number und FLOAT
Indizierung	In „Linien_geo“ nur für programminterne Selektionen	In eigener Oracle-spezifischer Tabelle „LINIEN_SDOINDEX“

Tabelle 6.1: Vergleich Topobase und GeoMedia Pro

Die Unterschiede der beiden Modelle liegen also in den Bezeichnungen der Tabellen und Tabellenspalten sowie in der Handhabung der räumlichen Indizierung. Diese hat bei der Topobase nichts mit dem räumlichen Index von Oracle zu tun, sondern wird von der Topobase-Struktur selbst angelegt und dient dem Modul TB_BasicMap dem schnelleren Zugriff auf den Datenbestand. Dieser Index wird als solcher nicht von Oracle außerhalb der Topobase-Softwareumgebung erkannt und benutzt.

Die Geometrien werden aber bei beiden Modellen in gleicher Weise abgelegt. Ist es also möglich, zumindest auf die Geometrien mit GeoMedia Professional zuzugreifen?

Dazu wird der Ablauf des Zugriffs zu Beginn generell beschrieben, da im weiteren Verlauf dieser Schritt noch des öfteren folgen wird.

Die einzelnen Datentöpfe, die man in GeoMedia Professional anbinden kann, nennt man „Warehouses“. Der sogenannte „Warehouse-Verbindungsassistent“ unterstützt den Nutzer bei der Herstellung einer Verbindung zu den Verschie-

denen Datenquellen. Die folgende Abbildung zeigt eine Auswahl der verschiedenen Datenserver, die jeweils eingebunden werden können.

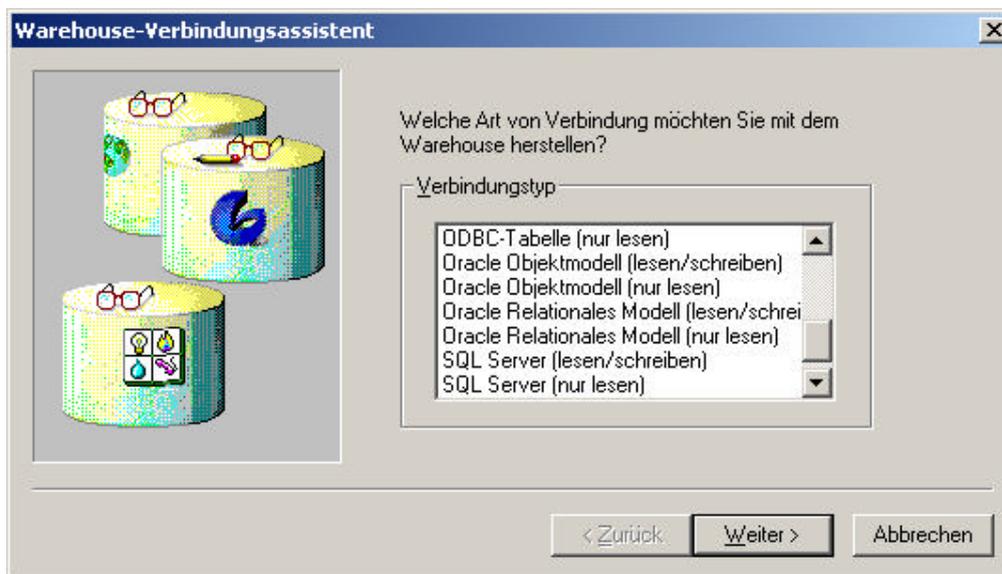


Abbildung 6-7: Warehouse-Verbindungsassistent

Für Oracle werden vier verschiedene Datenserver zur Verfügung gestellt. Diese beziehen sich auf das relationale und das objektrelationale Modell, mit jeweils nur lesendenden oder zusätzlich schreibenden Zugriff. Hier wird zunächst ein lesender Zugriff auf das relationale Modell ausgewählt, da noch keinerlei Metadaten vorhanden sind.

Im nachfolgenden Fenster müssen die Parameter für die Datenbankverbindung eingegeben werden.



Abbildung 6-8: Parameter für Datenbankverbindung

In weiteren Fenstern werden noch zusätzliche Eigenschaften der Verbindung abgefragt, die an dieser Stelle nicht weiter von Bedeutung sein sollen.

Ist eine Verbindung zu einem Warehouse hergestellt, können die verschiedenen Geometrietabellen im Kartenfenster nur dann angezeigt werden, wenn sie als sogenannte „Objektklasse“ in die Legende eingefügt worden sind.

Der Menüeintrag „Legende/Objektklasse hinzufügen ...“ führt zu folgendem Fenster, in dem die zu Verfügung stehenden Objektklassen zur Ansicht ausgewählt werden können.

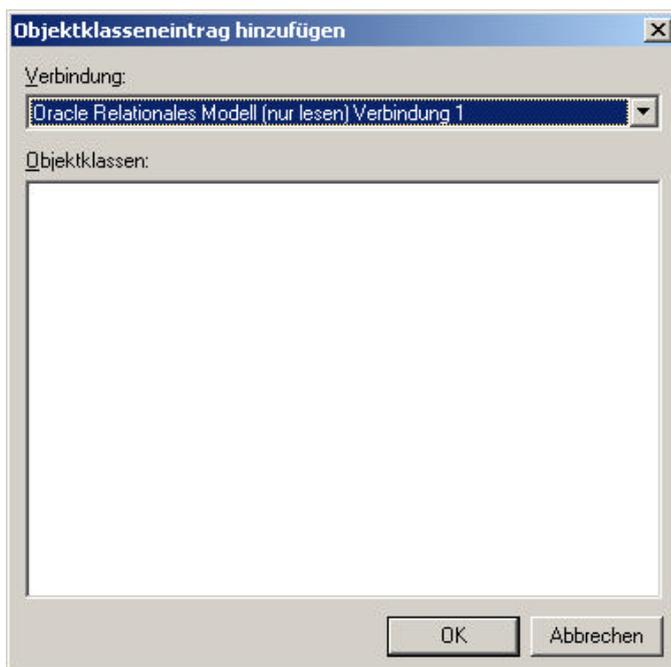


Abbildung 6-9: Objektklasseneintrag hinzufügen

Allerdings ist zu erkennen, dass keine Objektklassen vorhanden sind. Im Ergebnis bedeutet das, dass GeoMedia Professional nicht in der Lage ist, die Geometrien in den Tabellenstrukturen auf Anhieb zu erkennen. Die Tabellen an sich werden erkannt, da man sich diese als einfache Sachdatentabellen anzeigen lassen kann.

Um die Unterschiede in den Datenmodellen zu minimieren, kann man die Tabellen und Tabellenspalten entsprechend umbenennen, was jedoch nicht zum gewünschten Erfolg führt. Ein weiterer Unterschied besteht in der räumlichen Indizierung und den Metadaten.

Es ist generell möglich, die benötigte Metadatenstruktur zu einem vorhandenen Datenbestand aufzubauen. Dabei ist es weniger problematisch, die Tabellen zu erstellen, sondern diese mit den entsprechenden Werten zu befüllen.

Das Erstellen aller benötigten Metatabellen in einem vorhandenen Datenbestand kann in der Art erfolgen, dass man eine Verbindung mit Schreibzugriff zu diesem Datenserver herstellt, und dann eine neue Objektklasse in dieser Verbindung erstellt. Dabei werden automatisch die Metatabellen angelegt. Jedoch beziehen sich dann die Inhalte der Metatabellen nur auf die neue Objektklasse, und nicht auf den gesamten Datenbestand der Verbindung. Durch manuelles Eintragen müssten die Metatabellen um die Werte für den restlichen Datenbestand ergänzt werden. Da dies allerdings kompliziert und aufwendig ist, wird von GeoMedia Professional, wie bereits erwähnt, das „GDOO-Package“ bereitgestellt. Mit diesem Tool kann man die serverseitigen Metatabellen erstellen, die Tabellen *_SDOINDEX und *_SDODIM befüllen und Datenbanktrigger erstellen.

Diese Prozedur zur Erstellung der Metatabellen ist unumgänglich, wenn keine anderen Möglichkeiten bestehen. Im konkreten Fall des Datenzugriffs auf die Topobase-Daten gibt es jedoch noch eine weitere Möglichkeit.

Das Programm TB_Administrator bietet eine Funktion an, die es erlaubt, die Geometrie eines Objektes in ein einziges Datenbankfeld zu schreiben. Diese Datenspeicherung ist identisch mit dem objektrelationalen Ansatz, der zuvor in Kapitel 3 beschrieben wurde. Demnach gibt es ein Datenbankfeld vom Feldtyp MDSYS.SDO_GEOMETRY, in dem die Geometrie gespeichert wird. Da GeoMedia Professional auch in der Lage ist, das objektrelationale Datenmodell von Oracle lesend und schreibend zu benutzen, und somit den Datentyp MDSYS.SDO_GEOMETRY verarbeiten kann, soll nun versucht werden, diese Datenspalte zu erzeugen und anschließend darauf zuzugreifen. Dieses Vorgehen setzt selbstverständlich voraus, dass die Software von G-Plan zur Verfügung stehen muss.

Zunächst muss man mit dem TB_Administrator eine Verbindung zu dem Datenbankuser herstellen, dessen Geometrietabellen modifiziert werden sollen. Es erscheint die Baumstruktur des gesamten Inhaltes des Datenbankusers.

Wählt man in der linken Hälfte der Ansicht einen Layer aus, erscheinen im rechten Teil die Felddefinitionen der zugehörigen Featuretabelle. Die rechte Maustaste lässt das unten dargestellte Menü erscheinen.

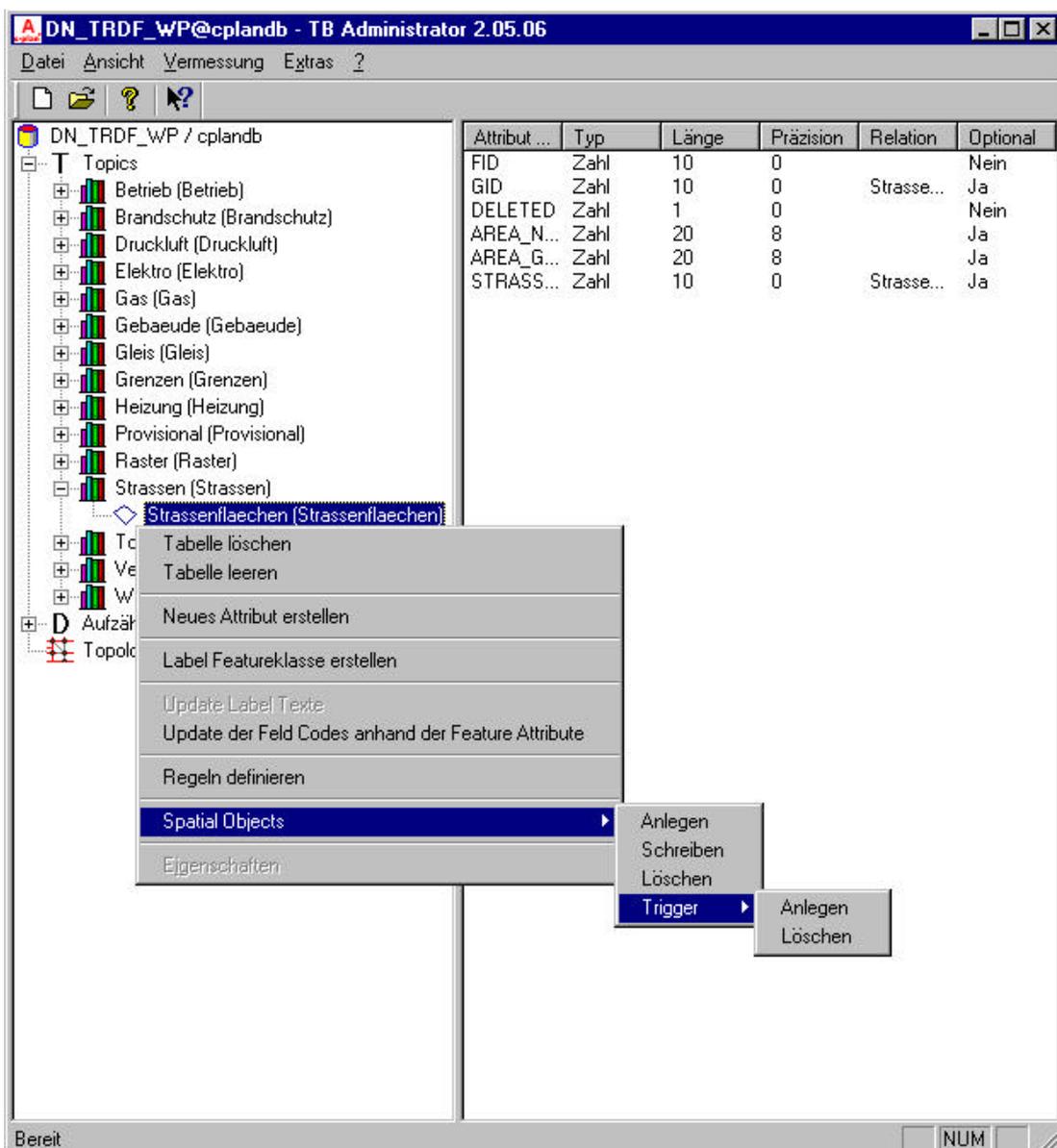


Abbildung 6-10: Definition der Spatial Objects

In diesem Menü kann man nun räumliche Objekte anlegen, schreiben oder löschen.

Zunächst muss man hier „Spatial Objects – Anlegen“ wählen. Durch diese Funktion wird ein Datenbankfeld in der Featuretabelle angelegt. Dieses Feld hat den Datentyp MDSYS.SDO_GEOMETRY und heißt „GEOMETRY“.

Damit das Feld und die späteren Einträge genauer definiert werden können, sind weitere Angaben erforderlich, die im sich automatisch zeigenden Fenster gemacht werden können. Diese Angaben werden für das in Kapitel 3 bereits beschriebene Statement

```
Create Type SDO_GEOMETRY As Object (
    SDO_GTYPE      Number,
    SDO_SRID       Number,
    SDO_POINT      MDSYS.SDO_PONT_TYPE,
    SDO_ELEM_INFO  MDSYS.SDO_ELEM_INFO_ARRAY,
    SDO_ORDINATES  MDSYS.SDO_ORDINATE_ARRAY);
```

benötigt.

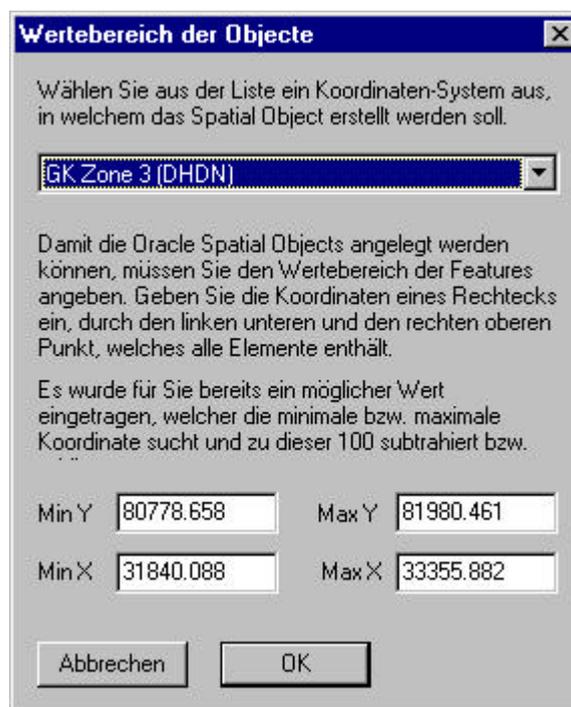


Abbildung 6-11: Wertebereich der Objekte

Wird dieses Fenster bestätigt, legt der TB_Administrator das Datenbankfeld an. Dieses Feld muss im nächsten Schritt noch befüllt werden. Dies kann mit der Funktion „Spatial Objects – Schreiben“ erfolgen. Dabei werden die Geometrieinformationen aus der <layername>_geo – Tabelle gelesen und in die Featuretable geschrieben. Die Zuordnung erfolgt über das Feld GID. Betrachtet man

sich nun die Featuretabelle eines Layers in der Oracle-Datenbank, stellt sich die neue Struktur folgendermaßen dar:

Tabelle: Standard Index-organisiert (IOT)

Spalten

Name	Schema	Datentyp	Größe	Nachk...	Ref	Null...	Standardwert
FID	<Kein>	NUMBER	10	0			
GID	<Kein>	NUMBER	10	0		✓	
DELETED	<Kein>	NUMBER	1	0			0
AREA_NOMINAL	<Kein>	NUMBER	20	8		✓	
AREA_GERECHNET	<Kein>	NUMBER	20	8		✓	
STRASSENART	<Kein>	NUMBER	10	0		✓	
GEOMETRY	MDSYS	SDO_GEOMETRY				✓	

Anwenden Wiederherstellen SQL zeigen Hilfe

Abbildung 6-12: Featuretabelle in Oracle

Das Vorhandensein einer Spalte, in der die Geometrie abgelegt ist, reicht alleine jedoch noch nicht aus, um mit GeoMedia Professional darauf zugreifen zu können. Eine Voraussetzung bei Fremddaten ist, dass die Metatabellenstruktur angelegt wird. Anders als beim relationalen Objektmodell aber bietet GeoMedia Professional hierbei ein Tool an, mit dem diese Struktur aufgebaut werden kann. Dieses „Datenbankdienstprogramm“ ist in der Programmgruppe zu GeoMedia Professional implementiert und kann dort gestartet werden. Der kurze Ablauf des Programms wird im Folgenden beschrieben.

Nach dem Start des Programms erfolgt die Auswahl des Datenbanktyps, für die eine Metadatenstruktur aufgebaut werden soll. Weiterhin müssen die Verbindungsparameter für diese Datenbank angegeben werden:



Abbildung 6-13: Datenbankverbindung

Anschließend erfolgt die Verbindungsaufnahme zu der Datenbank. Es wird ein Menü angezeigt, mit dem man Metadaten erstellen, Metadaten löschen, und weitere Funktionen durchführen kann.



Abbildung 6-14: Datenbank-Dienstprogramm

Wählt man nun „Objektklassen-Metadaten einfügen“, kann man im folgenden Fenster die Objektklassen auswählen, die in das Metadatenmodell mit aufgenommen werden sollen.

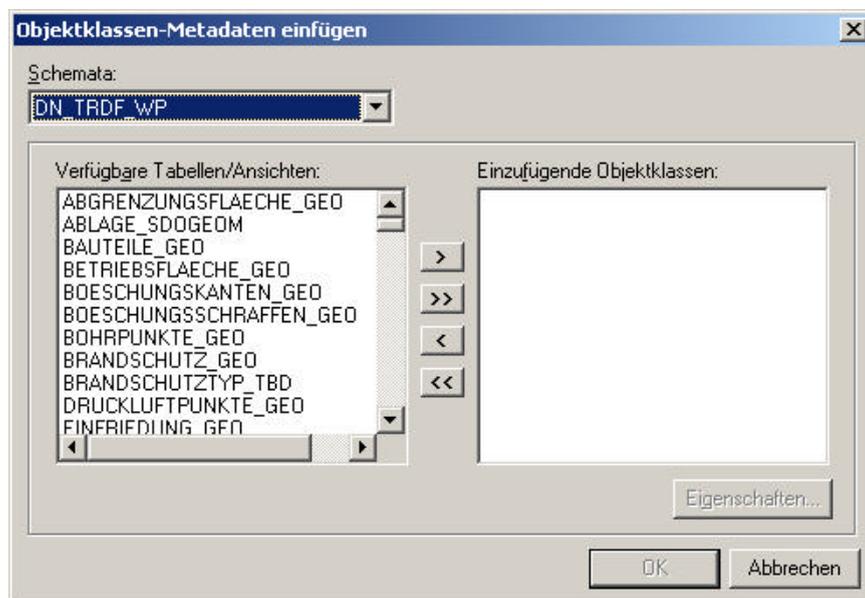


Abbildung 6-15: Objektklassen-Metadaten einfügen

Diese Schritte sind zunächst ausreichend, um sich die Daten in GeoMedia Professional anzeigen zu lassen.

Wählt man nun bei einer bestehenden Warehouse-Verbindung den Menüeintrag „Legende/Objektklasse hinzufügen ...“, erscheint ein mit vorhandenen und selektierbaren Objektklassen gefülltes Fenster.

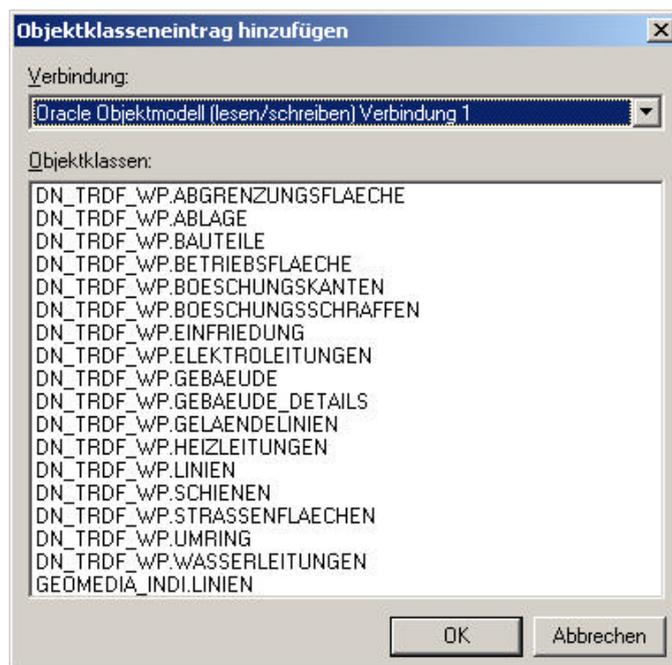


Abbildung 6-16: Objektklassenauswahl

Zu diesem Zeitpunkt haben die Daten noch keinen räumlichen Index. Daher können verschiedene Funktionalitäten von GeoMedia Professional, die eine räumliche Indizierung der Daten voraussetzen, nicht genutzt werden.

Dieser räumliche Index muss in der Oracle Datenbank selbst erstellt werden. Er kann entweder mit Hilfe des Oracle DBA-Studios, oder mit dem SQL-Worksheet angelegt werden. Er wird für jede Objektklasse separat definiert. Ist man im SQL-Worksheet mit dem entsprechenden Benutzer verbunden, kann das folgende Statement direkt eingegeben werden, um den Index zu erzeugen:

```
CREATE INDEX "DN_TRDF_WP"."INDEX_LINIEN"  
ON "DN_TRDF_WP"."LINIEN"("GEOMETRY") INDEXTYPE IS  
MDSYS.SPATIAL_INDEX PARAMETERS (' SDO_LEVEL=7  
SDO_COMMIT_INTERVAL=-1')
```

Dieses Statement besagt, dass ein Index im User „DN_TRDF_WP“ mit dem Namen „INDEX_LINIEN“ auf das Feld „GEOMETRY“ in der Tabelle „LINIEN“ erstellt werden soll, wobei die Kachelungstiefe gleich 7 sein soll.

Das Ergebnis der räumlichen Indizierung kann man sich in einem Zusatzprogramm zu Oracle Spatial anschauen, dem „Oracle Spatial Index Analyzer“. Dieses Tool eignet sich sehr gut zum Betrachten und Analysieren des räumlichen Indexes. Dabei bietet das Tool gewissermaßen Mapping-Funktionalitäten. Es sind Zoom-Werkzeuge und Möglichkeiten vorhanden, die die Darstellung der einzelnen Layer variieren lassen.

Die folgende Abbildung zeigt die Oberfläche des Index Analyzer und eine Darstellung der indizierten Gebäude mit der entsprechend gewählten Kachelung.

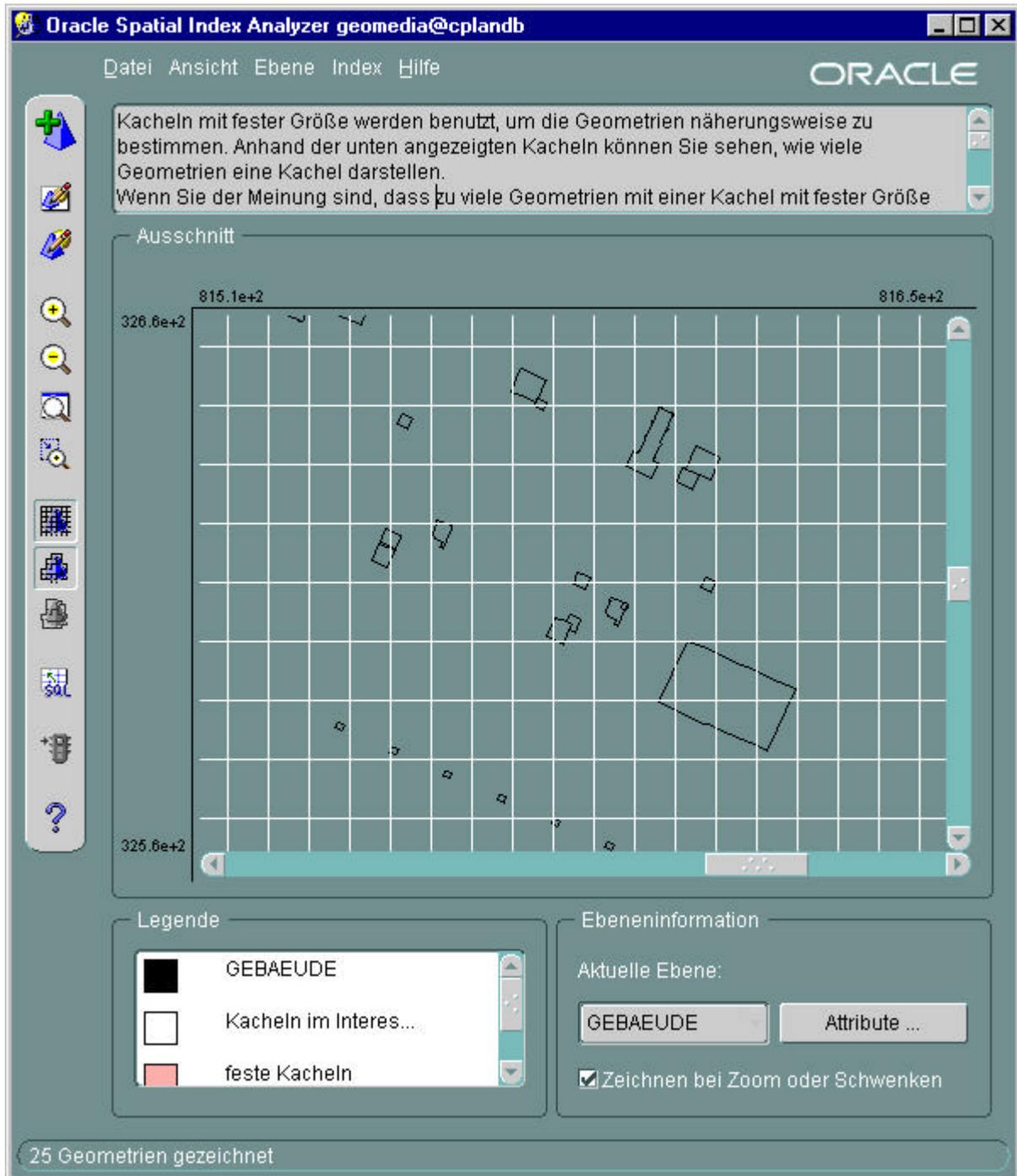


Abbildung 6-17: Oracle Spatial Analyzer

Da nun der gesamte Datenbestand sowohl Metadaten besitzt als auch räumlich indiziert wurde, kann nun auf die von der Topobase abgelegten Daten mit GeoMedia Professional zugegriffen werden, wobei es möglich ist, die räumlichen Analyse- und Abfragefunktionalitäten von GeoMedia Professional auf diesen Datenbestand anzuwenden.

Eine Darstellung der Daten, wie sie bereits in den beiden C-Plan - Systemen verwendet wurde, zeigt einen gesamten Standort mit Gebäuden, Strassen, Gleisanlagen, Böschungen und Umgrenzungen.

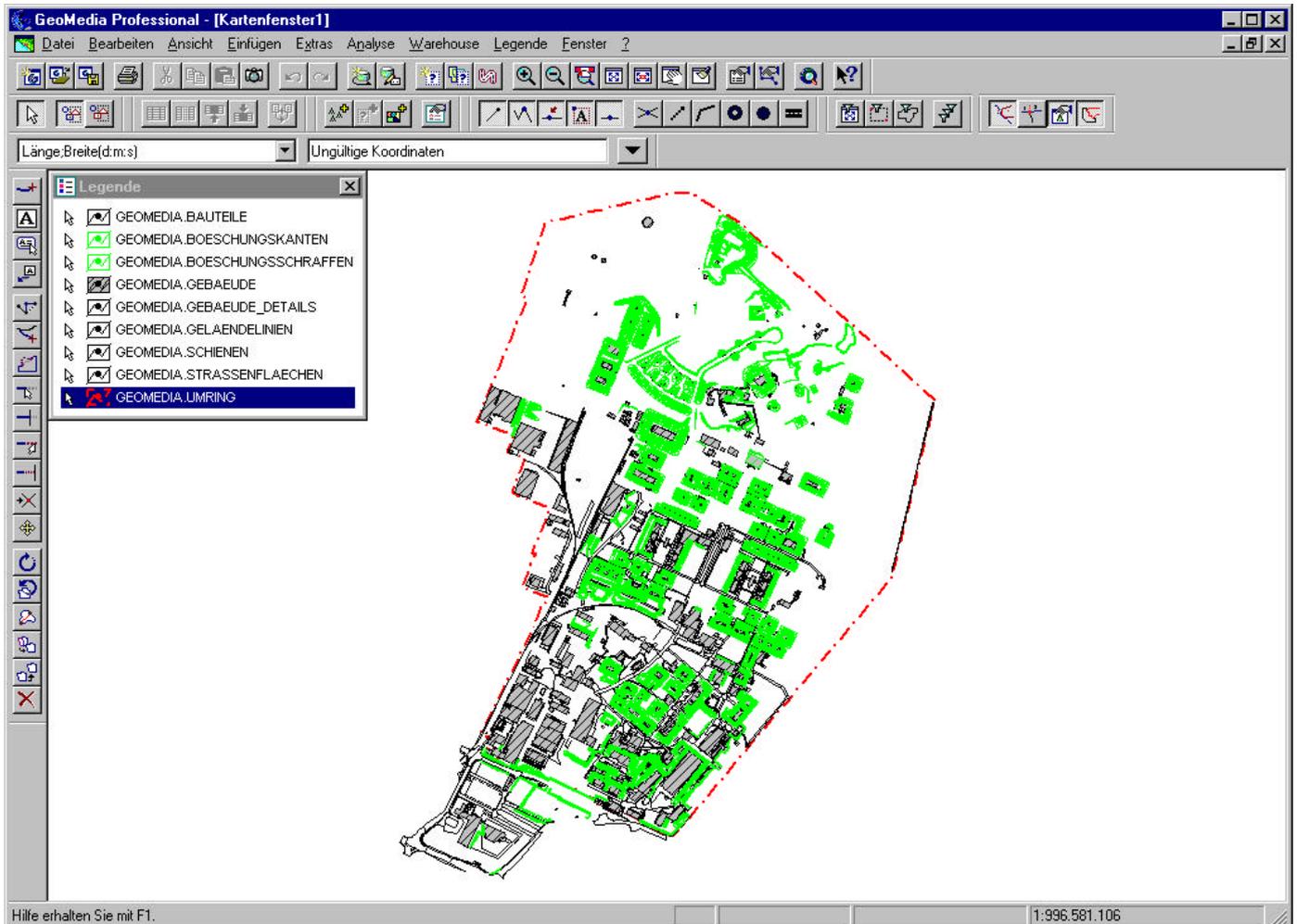


Abbildung 6-18: Karte aus Oracledatenbank in GeoMedia Professional

Während der gesamten Beschreibung der Abläufe wurde bewusst nur auf die Problematiken und die Lösungen des Datenzugriffs bezogen auf die Geometrien eingegangen. Selbstverständlich gibt es darüber hinaus noch eine Vielzahl an Sachdaten, die etwas in den Hintergrund geraten zu sein scheinen. Diese Informationen müssen aber durchaus nicht ungenutzt bleiben, oder gar verloren gehen. Da man nun einen Vollzugriff auf die Datenbank hat, ist es problemlos möglich, auch diese Daten mit GeoMedia Professional anzusprechen und zu nutzen. Da nun das objektrelationale Modell verwendet wird, liegen die neuen Geometrie- und die bereits vorhandenen Sachdaten in einer einzigen Tabelle

vor. Von daher sind die Attributinformationen, die mit den Geometrien verbunden waren, gleichermaßen vorhanden.

Lagen in der Topobase noch weitere Informationen in verbundenen Tabellen vor, kann ein Fremdsystem diese Strukturen nicht ohne weiteres erkennen und verarbeiten, da die Definitionen dieser Tabellenverknüpfungen meistens in weiteren Tabellen vorliegt, die von Topobase erstellt worden sind. Dennoch ist es aber durchaus machbar, innerhalb von GeoMedia Professional durch die Bildung von entsprechenden Joins weitere Datentabellen mit der Tabelle für die Objektklasse zu verbinden. Es ist beispielsweise sinnvoll, die Feature- und die Geometrietabelle der Topobase miteinander in GeoMedia Professional durch einen Join zu verbinden, damit auch diese Informationen direkt zur Verfügung stehen.

Solche Verbindungen lassen sich auch in Oracle selber realisieren, indem man mit Views arbeitet. Demnach könnte man einen View auf die beiden Tabellen „Linien“ und „Linien_geo“ erstellen und diesen View in GeoMedia Professional visualisieren. Damit würde man ebenfalls erreichen, dass man Zugriff auf beide Tabellen hat, wobei man durch diese Variante den Client, also GeoMedia Professional, von der Performanz her betrachtet, entlasten könnte.

Die letzte Frage, ob sich das System nun als interoperabel bezeichnen lässt, ist relativ schnell beantwortet. Laut Definition müsste es u.a. möglich sein, dass beide Systeme – C-Plan Topobase und INTERGRAPH GeoMedia Professional – die Daten in ein und der selben Datenbank ablegen. Daraus folgend sollte es möglich sein, mit beiden Systemen durch standardisierte Abfrage- und Analysealgorithmen auf den einen Datenbestand zuzugreifen.

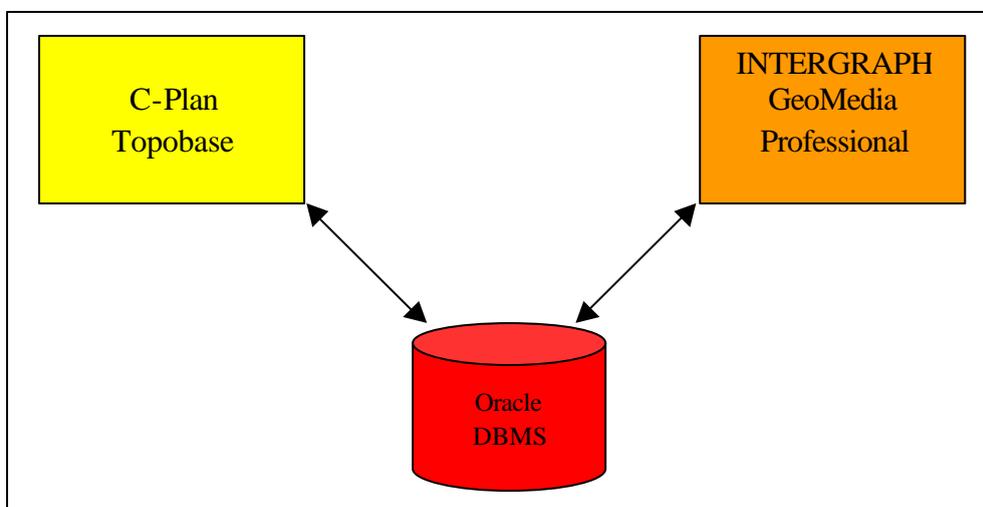


Abbildung 6-19: Interoperabler Datenzugriff

Dieses Kapitel hat gezeigt, dass es möglich ist, mit GeoMedia Professional auf die von der Topobase abgelegten Daten zuzugreifen, wenn die Geometrien zuvor mit Hilfe des TB_Administrators in ein einziges Feld vom Typ MDSYS.SDO_GEOMETRY abgelegt werden. Dieses Feld wird im objektrelationalen Modell von Oracle verwendet und wird auch von GeoMedia Professional zur Ablage der Geometrien verwendet. Dieses Geometriefeld wird von GeoMedia Professional auch angesprochen, wenn räumliche Abfragen und Analysen durchgeführt werden. Die Topobase aber hingegen spricht dieses Feld nicht bei solchen Funktionen an, sondern verwendet die Geometrieinformationen der standardmäßigen Dateiablage der Topobase, die in den <layer>_geo-Tabellen nach dem relationalen Modell stattfindet. Daraus folgt, dass zwar ein Zugriff auf Topobase-Daten von GeoMedia Professional aus möglich ist, aber nicht umgekehrt.

Weitaus problematischer gestaltet sich der Vorgang, wenn neue Tabellen in den jeweiligen Systemen definiert werden. Diese neuen Tabellen werden standardmäßig keinesfalls vom jeweils anderen System erkannt. Das liegt daran, dass beide Systeme mehr oder weniger umfangreiche Metadaten benötigen, in denen u.a. die Tabellen, Spaltendefinitionen und Beziehungen zwischen Tabellen festgehalten sind. Wenn nun beispielsweise in der Topobase eine neue Tabelle erzeugt wird, werden die Metadaten der Topobase automatisch aktualisiert. Um Zugriff auf diese Tabelle von GeoMedia Professional zu bekommen, muss diese Tabelle in die Metadatenstruktur von GeoMedia Professional aufgenommen werden, was keineswegs automatisch nur durch das Vorhandensein dieser Tabelle geschieht. Entweder muss die Tabelle von Hand in die Metadatenstruktur eingearbeitet, oder mit Hilfe des Datenbankdienstprogrammes aufgenommen werden. Der umgekehrte Fall äußert sich in gleicher Weise. Eine von GeoMedia Professional erstellte Tabelle wird nicht automatisch von der Topobase erkannt.

6.3 Zusammenfassung

Die umgesetzten Daten sollen neben ihrer Verwendung als Planungsgrundlage im Wesentlichen noch zwei weiteren Verwendungen zugeführt werden. Zum einen dienen sie als originärer Nutzen der Grundlage für Auskunftsarbeitsplätze, welche auf der Basis von MapGuide realisiert werden sollen. Auf der anderen Seite sollte es möglich sein, mit Drittsystemen auf den Datenbestand zuzugreifen, was dadurch getestet wird, indem exemplarisch mit der Software GeoMedia Professional von Intergraph der Datenbestand angesprochen wird.

Die Erstellung eines Auskunftssystems gestaltet sich als relativ unproblematisch, da die Firma C-Plan für diesen Zweck bereits fertige Module anbietet. Um die Abfragemasken und Menüs, die der Nutzer bereits von TB_Generic her kennt, auch im Web zur Verfügung zu haben, wird das Modul TB_GenericWeb eingesetzt. Dieses Modul arbeitet auf der Ebene des InternetInformation Servers und stellt die TB-Generic-Menüs und -Formulare als HTML-Dokumente dar. Eine benutzerdefinierte Anpassung der Web-Oberfläche ist durch herkömmliche ASP-, bzw. HTML-Programmierung möglich.

Der Zugriff auf den Datenbestand durch Drittsysteme gestaltet sich hingegen problematischer. Zum einen müssen für beide Systeme völlig verschiedene Metadatenstrukturen vorhanden sein, und selbst dann ist nicht gewährleistet, dass man uneingeschränkt auf die gegenseitigen Daten zugreifen kann. Ähnliche Problematiken treten auch auf, wenn solche Testreihen mit anderen GI-Systemen, wie z.B. MapInfo durchgeführt werden, da auch hier eine fest vorgegebene Metadatenstruktur verwendet wird.

Wie die Problematik der Metainformationen gelöst werden könnte, wird im abschließenden Kapitel angeregt. Das Problem der Geometrieablage im konkreten Beispiel der Topobase wird sich möglicherweise mit dem Erscheinen der Version 3 lösen lassen. In dieser Version soll ebenfalls das Objektmodell von Oracle verwendet werden. Dadurch wird auch in der Topobase die Geometrie in einem einzigen Datenfeld abgelegt werden. Das bedeutet, dass auch bei Analysen und Abfragen dieses Feld genauso wie bei GeoMedia Professional verwendet wird. Dann könnte es auch möglich werden, ein von einem Fremdsystem befülltes Geometriefeld anzusprechen.

In diesem Kapitel wurde bewusst der Schwerpunkt auf die Geometrieinformationen gelegt. Weitere Aspekte wie grafische Ausgestaltung, Koordinatensysteme oder Texte wurden nicht berücksichtigt. In der Topobase werden die Definitionen der grafischen Ausgestaltung in dem sogenannten Darstellungsmodell festgehalten. Dabei handelt es sich um drei miteinander verknüpfte Tabellen, in denen die Ausgestaltung für jede Featureklasse definiert werden kann. Dabei ist es durchaus möglich, attributabhängige Darstellungen festzulegen. Das System GeoMedia Professional verwaltet die grafische Ausgestaltung im GeoWorkspace. Dieser entspricht etwa der Projektdatei bei ArcView, oder einem Arbeitsbereich bei MapInfo. Aus dieser Sicht ist es kaum sinnvoll, auch die Definition der grafischen Ausgestaltung zu migrieren, da diese in den meisten Systemen nicht zusammen mit den Daten gespeichert werden.

7 Abschlussbewertung und Ausblick

Im letzten Kapitel soll der gesamte Ablauf der Arbeit noch einmal kurz aufgegriffen werden, um eine abschließende Betrachtung der Ergebnisse zuzulassen.

Die beschriebene Migration des Datenbestandes hat etwa einen Zeitraum von einem dreiviertel Jahr in Anspruch genommen. Einen sehr großen Anteil an dieser Zeit hat die Sichtung und das Verstehen der alten Datenstruktur. Es hat sich als sehr aufwendig erwiesen, den historisch gewachsenen Datenbestand zu analysieren. Immer wieder stieß man dabei an Fragestellungen, deren Beantwortung nur durch eine gute Kenntnis der damals aktuellen und üblichen Datenmodellierungsansätze möglich gewesen ist. Zahlreiche Zwangsbedingungen, die es in heutigen Datenbanksystemen nicht mehr gibt, ließen nur eine recht umständliche Datenhaltung zu. Beim Analysieren des alten Datenbestandes musste sich immer wieder gefragt werden, wie eine entdeckte Teilstruktur nach heutigen Gesichtspunkten und Möglichkeiten abgelegt werden könnte. Auch galt es stets zu entscheiden, welche Informationen bei einer Migration unter Umständen nicht mehr benötigt würden, und welche auf keinen Fall verloren gehen dürfen. Daher hat es sich als sehr sinnvoll ergeben, die Daten Stück für Stück umzusetzen und nicht etwa in einem Arbeitsschritt. Das Befüllen der Konvertierungstabellen hat bei der technischen Umsetzung die meiste Zeit in Anspruch genommen, da hierbei in verschiedenen SQL-Statements die Strukturen der alten und der neuen Datenbank eingearbeitet und berücksichtigt werden mussten.

Im Endeffekt kann man aber sagen, dass sich der Aufwand mit Sicherheit gelohnt hat. Neben einer nie zuvor dagewesenen Kenntnis des Datenbestandes hat dieser durch die Migration sicherlich erheblich an Wert gewonnen. Durch die Integration in eine neue Softwareumgebung sind nun die Funktionalitäten eines GIS bezogen auf Analyse- und Verschneidungsfunktionen möglich. Es können Entscheidungen, die von baulichen Maßnahmen, über Besitz- und somit bis hin zu Abrechnungstechnischen Fragestellungen reichen, schneller beantwortet, oder bearbeitet werden. Über dieses Mehr an Leistungsfähigkeit haben sich die nicht unerheblichen Kosten der Datenmigration schon bald gerechnet.

Das gesteckte Ziel, den Datenbestand durch die IMS-Technologie an verschiedene Arbeitsplätze zu verteilen, ist mit Sicherheit erreicht worden. Es ist dadurch möglich, die Daten an der Stelle zu pflegen, an der sie für planerische Zwecke auch benötigt werden. Das Ingenieurbüro bietet das technische Wissen, um die vom jeweiligen Konzernmitarbeiter benötigten Informationen online zur Verfügung zu stellen. Daher ist es nicht mehr notwendig, das Ingenieurbüro wegen jeder Auskunft per Telefon oder Email zu bemühen, wodurch nicht nur Zeit, sondern auch Kosten gespart werden. Ein weiterer Vorteil besteht darin, dass in Schadens-, oder Unglücksfällen, was bei einem Konzern dieser Größenordnung nicht selten vorkommt, wichtige Informationen, wie etwa Stauraumvolumen von Kanälen für abfließendes Löschwasser, oder Leitungsverläufe im Gas- oder Stromnetz, zu jeder Tages- und Nachtzeit abgerufen werden können, wodurch möglicherweise größere Schäden verhindert werden können.

Die Verwendung der beschriebenen Module lässt eine sehr detaillierte Anpassung an die Bedürfnisse des Konzerns zu. Mit Standardprogrammierungsumgebungen, wie ASP, Visual Basic oder HTML lassen sich jederzeit Ergänzungen oder Veränderungen durchführen.

Das weitere Ziel, eine offene und standardisierte Datenhaltung zu gewinnen, ist nach derzeit aktuellen Gesichtspunkten als durchaus erfüllt zu beurteilen. Zwar hat sich gezeigt, dass die Systeme keineswegs als vollkommen interoperabel zu bezeichnen ist, aber fest steht, dass ein verlustfreier Datenaustausch mit anderen Systemen möglich ist. Auch der Zugriff auf den Datenbestand mit einem Fremdsystem ist ebenfalls als gelungen und möglich zu erachten. Das Entscheidende war, dass kein Datenverlust beim Zugriff auftritt. Da immer mehr Systemhersteller die standardisierte Datenhaltung von Oracle unterstützen, ist auch ein Datenaustausch zwischen den Systemen möglich. Durch verschiedenste Schnittstellen ist es möglich zwischen den Systemen über Fremdformate zu korrespondieren.

Wenn man nun die im Titel etwas provokante Frage nach dem Standard aufgreift, kommt man zu der Feststellung, dass die Systeme von einer angestrebten Interoperabilität noch weit entfernt sind. Auch wenn die Ablage der Daten standardisiert in einem objektorientierten Oracle-Umfeld erfolgt, sind doch noch eine ganze Reihe von Metadaten für die Systeme erforderlich, um den Datenbestand anzusprechen. Der eine Weg wäre also, eben diese Metadatenstrukt-

ren für alle Systeme einheitlich zu gestalten, so dass kein Abgleich mehr erforderlich würde. Dadurch würden sich zwar die verschiedenen Systeme immer mehr ähneln, aber es wäre ein großer Schritt in Richtung Interoperabilität gemacht. Eine weitere Möglichkeit besteht darin, eigene Applikationen zu entwickeln, welche die Metadatenstrukturen stets aktualisieren. Solche Applikationen zu erstellen gestaltet sich technisch nicht zu schwer, da die Metadatenstrukturen durchaus offengelegt sind, oder in den konkreten Beispielen nicht sonderlich schwer zu durchschauen sind. Es gibt also hinreichende Metainformationen über die Metadatenstrukturen. Wenn nun die beiden beschriebenen Systeme parallel über dem selben Datenbestand verfügen sollen, müsste dieser Datenbestand zum einen die Metadaten beider Systeme beinhalten, und zum anderen eine Applikation zwischengeschaltet werden, welche die Metadaten stets um neue Informationen ergänzt. Dieser Ansatz einer zusätzlichen Applikation wurde bereits von Blankenbach (2001) mit der Erstellung eines Metadatenmanagers realisiert.

8 Literaturverzeichnis

Dynamit Nobel GmbH: Homepage des Konzerns, <http://www.dynamit-nobel.com> , (02/11/24)

C-Plan (a), (1999): Handbuch Topodat, (15-19)

Oracle (2000): Oracle Spatial User's Guide and Reference, Release 8.1.7, 2000

Bärk (2002): Datenaustausch zwischen Geoinformationssystemen, <http://ddgi.gfz-potsdam.de/vorgis95.html> , (28.12.2002)

CISS (2002): Homepage der Firma, <http://www.ciss.de> , (28.12.2002)

SAFE (2002): Homepage der Firma, <http://www.safe.com> , (28.12.2002)

Buhmann/Wiesel (2002): GIS-Report, Bernhard Harzer Verlag GmbH, Karlsruhe, (287)

C-Plan (b), (2002): Homepage der Firma, <http://www.cplan.de/home.htm> , (02/12/04)

C-Plan (c), (2002): Topobase Grundlagen, (3-6)

Autodesk, (2002): Benutzerhandbuch MapGuide

Bill, (2001): Lexikon der Geoinformatik, Herbert Wichmann Verlag, Heidelberg, (141)

Blankenbach, (2001): Umsetzung und Grenzen der Interoperabilität zwischen vier ausgewählten GI-Systemen auf der Basis von Oracle 8i Spatial, Diplomarbeit, unveröffentlicht

Anhang A

Die Tabellen der Topobase-Grundstruktur:

TB_ATTRIBUTE

TB_ATTRIBUTE_MAPPING

TB_CENTROID_RELATIONS_IN_MUT

TB_CONSTRAINT

TB_DICTIONARY

TB_DIMENSION

TB_DIMENSION_RELATION

TB_DISPLAY_MODEL

TB_DISPLAY_MODEL_DEFINITION

TB_DISPLAY_MODEL_PRIORITY

TB_DOCUMENT

TB_ENUMS

TB_EVENT_LOG

TB_EVENT_MAPPING

TB_FEATURES_IN_MUTATION

TB_GEOLINE_IMPORT

TB_INFO

TB_INSERT_FEATURE

TB_INTERLIS_AREAS

TB_INTERLIS_LABELS

TB_INTERLIS_MAPPING

TB_INTERLIS_SYMBOLS

TB_LAENGSSCHNITT_DEFINITION

TB_LAENGSSCHNITT_DETAIL

TB_LAENGSSCHNITT_SUB_DETAIL

TB_LANGUAGE

TB_MUTATION

TB_POINT_NUMBER

TB_POSITION_FINDER

TB_RANGES

TB_RELATIONS
TB_REPORT_CONSTRUC
TB_REPORT_POINT
TB_RULE_BASE
TB_TOPOLOGY
TB_TOPOLOGY_DEFINITION
TB_TOPOLOGY_STATISTICS_REPORT
TB_VARIABLE
TB_VERSCHSTAP
TB_VERSCHSTAP_ALT_NEU
TB_VERSCHSTAP_DEFINITION
TB_VERSCHSTAP_MUTATION
TB_WORKSPACE
TB_WORKSPACE_CATEGORY
TB_WORKSPACE_DEFINITION
TB_WS_DATATYPE
TB_WS_INTERFACE
TB_WS_INTERFACETYPE
TB_WS_INTERFACE_DEFINITION
TB_WS_POSTPROCESSING
TB_WS_PREPROCESSING

Anhang B

Die selbstdefinierten Tabellen eines gesamten Standortes (ohne Kanal):

ABGRENZUNGSFLAECHE
ABGRENZUNGSFLAECHE_GEO
ABLAGE
ABLAGE_GEO
BAUTEILE
BAUTEILE_GEO
BETRIEBSFLAECHE
BETRIEBSFLAECHE_GEO
BOESCHUNGSKANTEN
BOESCHUNGSKANTEN_GEO
BOESCHUNGSSCHRAFFEN
BOESCHUNGSSCHRAFFEN_GEO
BOHRPUNKTE
BOHRPUNKTE_GEO
BRANDSCHUTZ
BRANDSCHUTZTYP_TBD
BRANDSCHUTZ_GEO
DRUCKLUFTPUNKTE
DRUCKLUFTPUNKTE_GEO
EIGENTUEMER
EINFRIEDUNG
EINFRIEDUNGSART_TBD
EINFRIEDUNG_GEO
ELEKTROANLAGEN
ELEKTROANLAGENTYP_TBD
ELEKTROANLAGEN_GEO
ELEKTROLEITUNGEN
ELEKTROLEITUNGEN_GEO
ELEKTROLEITUNGSART_TBD
ELEKTROPUNKTE

ELEKTROPUNKTETYP_TBD
ELEKTROPUNKTE_GEO
ELEKTROTRASSE
ELEKTROTRASSE_GEO
ELEKTROTRASSE_TBL
ELEKTROTRASSE_TBL_GEO
ELEKTROVERTEILER
ELEKTROVERTEILERART_TBD
ELEKTROVERTEILER_GEO
FLURSTUECKE
FLURSTUECKE_GEO
FLURSTUECKE_TBA
FLURSTUECKE_TBA_GEO
GASLEITUNGEN
GASLEITUNGEN_GEO
GASPUNKTE
GASPUNKTE_GEO
GEBAEUDE
GEBAEUDEPUNKTE
GEBAEUDEPUNKTE_GEO
GEBAEUDE_DETAILS
GEBAEUDE_DETAILS_GEO
GEBAEUDE_GEO
GEBAEUDE_TBL
GEBAEUDE_TBL_GEO
GEB_FUNKTION_TBD
GEB_LINIEN_ART_TBD
GELAENDELINIEN
GELAENDELINIEN_ARTEN_TBD
GELAENDELINIEN_GEO
GLEISPUNKTE
GLEISPUNKTE_GEO
GRENZPUNKTE
GRENZPUNKTE_GEO

HEIZLEITUNGEN
HEIZLEITUNGEN_GEO
HEIZLEITUNGSART_TBD
HEIZPUNKTE
HEIZPUNKTE_GEO
LANDESGRENZE
LANDESGRENZE_GEO
LINIEN
LINIEN_GEO
PLANUNGEN
PLANUNGEN_GEO
PUNKTE
PUNKTE_GEO
PUNKTE_TBL
PUNKTE_TBL_GEO
RASTERLINIEN
RASTERLINIEN_GEO
SAVEB
SAVEB_GEO
SAVEB_TBL
SAVEB_TBL_GEO
SCHIENEN
SCHIENEN_GEO
STRASSENART_TBD
STRASSENFLAECHE
STRASSENFLAECHE_GEO
TOPPUNKTART_TBD
TOPPUNKTE
TOPPUNKTE_GEO
UMRING
UMRING_GEO
WASSERARMATUREN
WASSERARMATURENTYP_TBD
WASSERARMATUREN_GEO

WASSERLEITUNGEN
WASSERLEITUNGEN_GEO
WASSERLEITUNGSART_TBD
WASSERLEITUNGSMATERIAL_TBD
WASSERPUNKTE
WASSERPUNKTE_GEO
WASSERSCHUTZPUNKTE
WASSERSCHUTZPUNKTE_GEO
WASSERSCHUTZZONE
WASSERSCHUTZZONE_GEO

Die selbstdefinierten Tabellen eines gesamten Standortes (nur Kanal):

AW_ABLAUFART_TBD
AW_ABWASSER_ART_TBD
AW_ALEITUNG_FKT_HIERAR_TBD
AW_ALEITUNG_MATERIAL_TBD
AW_APUNKT_ART_TBD
AW_APUNKT_FUNKTION_TBD
AW_APUNKT_MATERIAL_TBD
AW_AUSLASTUNG_TBD
AW_BEGR_BESCHICHTUNG_TBD
AW_BERECHNUNG_HYD
AW_BESCHICHTUNG_AUSSEN_TBD
AW_BESCHICHTUNG_INNEN_TBD
AW_BEWERTUNGSFAKTOR_H_TBD
AW_BEWERTUNGSFAKTOR_Q_TBD
AW_BILD_MEDIUM_TBD
AW_BODENART_TBD
AW_CHANGED_COLUMNS
AW_DETAIL_FUNKTION_TBD
AW_DETAIL_PUNKT_ART_TBD
AW_DETAIL_PUNKT_FKT_TBD
AW_DICHTUNGSART_TBD

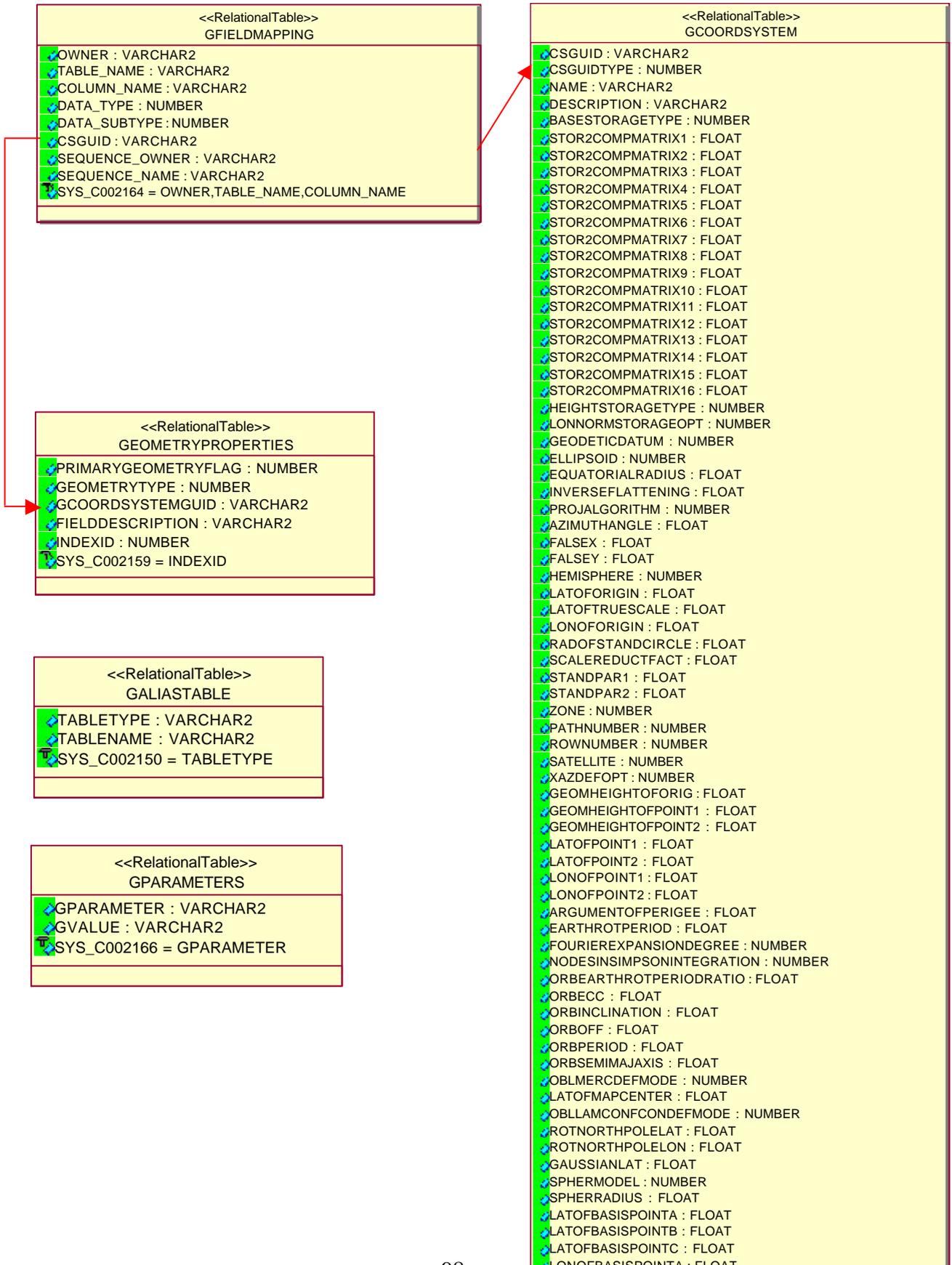
AW_GRUNDWASSERABSTAND_TBD
AW_HALTUNG
AW_HALTUNG_BETTUNG_TBD
AW_HALTUNG_FKT_HIERAR_TBD
AW_HALTUNG_FKT_HYDRAU_TBD
AW_HALTUNG_GEO
AW_HALTUNG_MATERIAL_TBD
AW_HALTUNG_PROFILEINBAU_TBD
AW_HALTUNG_TBL
AW_HALTUNG_TBL_GEO
AW_HALTUNG_VERB_ART_TBD
AW_HOEHGENAUIGKEIT_TBD
AW_HYD_BEMESSUNGSREGEN_TBD
AW_HYD_REGEN_UEB_ART_TBD
AW_HYD_SBWART_AUSLASS_F_TBD
AW_HYD_SBWART_AUSLASS_TBD
AW_HYD_SBWART_PUMPE_TBD
AW_HYD_SBWART_SCHIEBER_TBD
AW_HYD_SBWART_WEHR_TBD
AW_HYD_VERFAHREN_TBD
AW_HYDRAULIK_HALTUNG
AW_IMPORT_SCHNITTST_BESCHR
AW_IMPORT_SCHNITTST_DEF
AW_INSP_H_ART_TBD
AW_INSP_HALTUNG
AW_INSP_KENNUNG
AW_INSP_KENNUNG_TMP
AW_INSP_KENNUNGSZUSTAND
AW_INSP_KNGZUSTAND_TMP
AW_INSP_KUERZELTYP_TBD
AW_INSP_L_ART_TBD
AW_INSP_RICHTUNG_TBD
AW_INSP_S_KENNUNG
AW_INSP_S_KENNUNGSZUSTAND

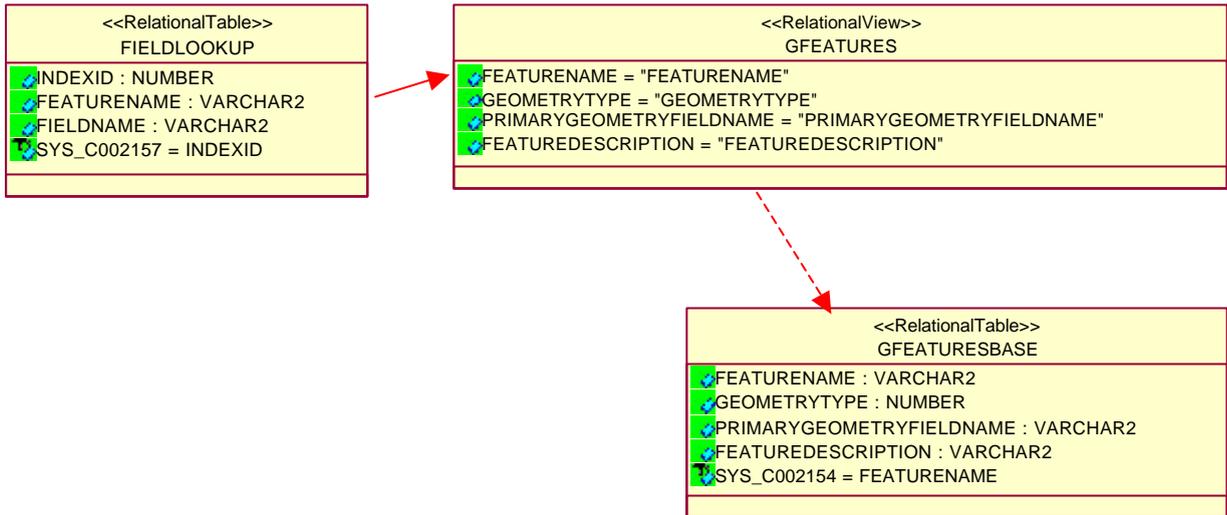
AW_INSP_S_TEXTZUSATZ_TBD
AW_INSP_STRECKENSCHADEN_TBD
AW_KANALART_TBD
AW_KLAS_FAKTOR
AW_KLAS_MODELL
AW_KLAS_RANDBEDINGUNG
AW_LAGEGENAUIGKEIT_TBD
AW_MODELLTYP_TBD
AW_NETZLINIE
AW_NETZLINIE_GEO
AW_NUTZUNGS_ART_TBD
AW_PROFILART_TBD
AW_SBW_RANDBEDINGUNG_TBD
AW_SBW_TRENN_WARN_TBD
AW_SBWART_DROSSEL_TBD
AW_SBWART_EINLEIT_TBD
AW_SBWART_KLAER_TBD
AW_SBWART_PUMPE_TBD
AW_SBWART_SCHIEBER_TBD
AW_SBWART_SPEICHER_TBD
AW_SBWART_TRENN_TBD
AW_SBWART_WEHR_TBD
AW_SCHACHT
AW_SCHACHT_ART_TBD
AW_SCHACHT_BAUGR_FORM_TBD
AW_SCHACHT_DART_TBD
AW_SCHACHT_DECKEL
AW_SCHACHT_DECKEL_GEO
AW_SCHACHT_DL_BELASTUNG_TBD
AW_SCHACHT_EINST_HILFE_TBD
AW_SCHACHT_FORM_TBD
AW_SCHACHT_FUNKTION_TBD
AW_SCHACHT_GEO
AW_SCHACHT_GERINNEFORM_TBD

AW_SCHACHT_MATERIAL_TBD
AW_SCHACHT_SOHLE
AW_SCHACHT_SOHLE_GEO
AW_SCHACHT_STEIGEISENG_TBD
AW_SCHACHT_UNTENFORM_TBD
AW_SCHACHTDECKELFORM_TBD
AW_SCHACHTDETAIL
AW_SCHACHTDETAIL_GEO
AW_SK_MEDIUM_TBD
AW_SK_TRASSE_ART_TBD
AW_SONDERBAUWERK_DART_TBD
AW_TRASSE_DART_TBD
AW_VERKEHRSFLAECHE_TBD
AW_VERLEGEART_TBD
AW_WASSERSCHUTZZONE_TBD
AW_WEHRKRONENART_TBD
AW_ZUGAENGLICHKEIT_TBD
AW_ZUSTAND_HALTUNG
AW_ZUSTANDSKLASSENTYP_TBD

Anhang C

Die Metatabellenstruktur des Schemas GDOSYS





<<RelationalTable>> MODIFICATIONLOG	
MODIFICATIONNUMBER : NUMBER	
TYPE : VARCHAR2	
MODIFIEDTABLEID : NUMBER	
SESSIONID : NUMBER	
MODIFIEDDATE : DATE	
KEYVALUE1 : VARCHAR2	
KEYVALUE2 : VARCHAR2	
KEYVALUE3 : VARCHAR2	
KEYVALUE4 : VARCHAR2	
KEYVALUE5 : VARCHAR2	
KEYVALUE6 : VARCHAR2	
KEYVALUE7 : VARCHAR2	
KEYVALUE8 : VARCHAR2	
KEYVALUE9 : VARCHAR2	
KEYVALUE10 : VARCHAR2	
KEYVALUE11 : VARCHAR2	
KEYVALUE12 : VARCHAR2	
KEYVALUE13 : VARCHAR2	
KEYVALUE14 : VARCHAR2	
KEYVALUE15 : VARCHAR2	
KEYVALUE16 : VARCHAR2	
SYS_C002152 = MODIFICATIONNUMBER	

<<RelationalTable>> MODIFIEDTABLES	
MODIFIEDTABLEID : NUMBER	
TABLENAME : VARCHAR2	
KEYFIELD1 : VARCHAR2	
KEYFIELD2 : VARCHAR2	
KEYFIELD3 : VARCHAR2	
KEYFIELD4 : VARCHAR2	
KEYFIELD5 : VARCHAR2	
KEYFIELD6 : VARCHAR2	
KEYFIELD7 : VARCHAR2	
KEYFIELD8 : VARCHAR2	
KEYFIELD9 : VARCHAR2	
KEYFIELD10 : VARCHAR2	
KEYFIELD11 : VARCHAR2	
KEYFIELD12 : VARCHAR2	
KEYFIELD13 : VARCHAR2	
KEYFIELD14 : VARCHAR2	
KEYFIELD15 : VARCHAR2	
KEYFIELD16 : VARCHAR2	
SYS_C002151 = MODIFIEDTABLEID	

<<RelationalTable>> ATTRIBUTEPROPERTIES	
ISKEYFIELD : NUMBER	
FIELDDESCRIPTION : VARCHAR2	
INDEXID : NUMBER	
FIELDFORMAT : VARCHAR2	
FIELDTYPE : NUMBER	
ISFIELDDISPLAYABLE : NUMBER	
FIELDPRECISION : NUMBER	
SYS_C002158 = INDEXID	

Folgende Tabellenspalten korrespondieren durch sich implizierende Beziehungen miteinander:

