



Master Thesis

im Rahmen des
Universitätslehrganges „Geographical Information Science & Systems“
(UNIGIS MSc) am Zentrum für GeoInformatik (Z_GIS)
der Paris Lodron-Universität Salzburg

zum Thema

„Interoperable Geodatennutzung in heterogenen Systemarchitekturen über direkte Datenbankzugriffe“

vorgelegt von

Dipl.-Ing. (FH) Lothar Göttmann
U1319, UNIGIS MSc Jahrgang 2007

Zur Erlangung des Grades
„Master of Science (Geographical Information Science & Systems) – MSc(GIS)“

Gutachter:
Ao. Univ. Prof. Dr. Josef Strobl

Ditzingen, 08.06.2009

Eidesstattliche Erklärung

„Ich versichere, diese Master Thesis selbstständig, ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit die wörtlich oder sinngemäß übernommen wurden, sind entsprechend gekennzeichnet.“

Ditzingen, Juni 2009

.....

(Lothar Göttmann)

Vorwort

Die vorliegende Master Thesis ist im Rahmen des Universitätslehrganges „Geographical Information Science & Systems“ (UNIGIS MSc) am Zentrum für GeoInformatik (Z_GIS) der Paris Lodron-Universität Salzburg zur Erlangung des Grades „Master of Science (Geographical Information Science & Systems) – MSc(GIS)“ entstanden.

Ich möchte mich bei Herrn Ao. Univ. Prof. Dr. Josef Strobl für die Leitung, Betreuung und die zielführenden und wertvollen Anregungen zu dieser Master Thesis ganz besonders bedanken.

Dem gesamten UNIGIS Lehrgangsteam, insbesondere Herrn Mag. Michael Fally und Frau Mag. Julia Moser, danke ich herzlichst für die interessanten, lehrreichen und spannenden Einblicke in die Geoinformationwissenschaften und die Unterstützung während meiner Studienzeit einschließlich dieser Arbeit.

Ich bedanke mich bei den Unternehmen Oracle Corporation, Environmental Systems Research Institute, Inc. und disy Informationssysteme GmbH für die Bereitstellung der im Rahmen dieser Arbeit verwendeten Software.

Mein größter Dank gilt meiner Familie für ihr Verständnis und die uneingeschränkte und ermutigende Unterstützung während meiner gesamten Studienzeit.

Lothar Göttmann

Ditzingen, Juni 2009

Zusammenfassung

In vielen Institutionen sowohl aus dem öffentlichen als auch aus dem privaten Sektor hat sich der Einsatz von Geodaten und Geoinformationssystemen zur Unterstützung der Geschäftsprozesse schon seit vielen Jahren etabliert. Zur Verwaltung und Speicherung der Geodaten werden dazu seit Ende der 1990er Jahre zunehmend Datenbanksysteme eingesetzt.

Im Zuge der allgemein gesteigerten Integration und Vernetzung von Informationen und Systemen gewinnt die Nutzung verteilter, heterogener Geodatenbanken in der Geoinformationsverarbeitung, gerade in mittleren und großen Organisationen, kontinuierlich an Bedeutung. Die Entwicklung und der Ausbau von unternehmensweiten und -übergreifenden GIS-Infrastrukturen zur interoperablen Geodatennutzung und deren Kopplung mit fachlich diversen IT-Bereichen schreitet stetig voran.

Im Kontext einer interoperablen Geodatennutzung nimmt der Zugriff auf die verteilten Geodatenbanken eine entscheidende Rolle ein. Im Rahmen dieser wissenschaftlichen Arbeit wird diesbezüglich der Ansatz der direkten Zugriffe von GIS-Anwendungen auf Geodatenbanksysteme ohne Zwischenschichten über die standardisierte Schnittstelle und Abfragesprache SQL verfolgt und hinsichtlich seiner technischen Qualifizierung und der strategischen Eignung in einer heterogenen Architektur untersucht. Die Betrachtung schließt dabei gleichermaßen die konzeptuellen, räumlichen Datenmodelle als auch die existierenden Technologien zur Realisierung mit ein.

Abstract

In many institutions both from the public and from the private sector the use of spatial data and geographic information systems to support the business processes have been established for many years. Since the late 1990s database systems are used increasingly to manage and store the spatial data.

In the course of a general increase in integration and networking of information and systems the use of distributed heterogeneous spatial databases in spatial data processing is gaining in importance steadily, especially in medium and large organizations. The development and expansion of company-wide and company-cross GIS infrastructures for interoperable spatial data use and their coupling with a variety of professional IT fields is progressing continuously.

In the context of an interoperable spatial data use the access to distributed spatial databases takes a crucial role. In this regard the approach of direct access from GIS applications to spatial database systems without intermediate layers using the standardized query language SQL is pursued, and its technical qualification and strategic applicability in a heterogeneous architecture is examined within the scope of this scientific work. The consideration encloses equally the conceptual spatial data models as well as the existing technologies for the realization.

Inhaltsverzeichnis

1 Einführung	1
1.1 Problemstellung und Motivation	1
1.2 Fragestellungen und Ziele	2
1.3 Lösungsansatz	4
1.3.1 Theorie	4
1.3.2 Methoden	5
1.3.3 Werkzeuge	5
1.3.4 Daten	5
1.4 Thematische Einschränkung	5
1.5 Struktur der Master Thesis	7
2 Grundlagen und Literaturanalyse	9
2.1 Literaturüberblick	9
2.2 Interoperabilität und Standardisierung	11
2.2.1 Interoperabilität	11
2.2.2 Standardisierung	13
2.3 Geodatenbanksysteme und Geodaten	16
2.3.1 Geodatenbanksysteme	16
2.3.2 Räumliche Datenmodelle und Datenbankschemata	25
2.4 Architekturen und Technologien	33
2.4.1 Softwarearchitekturen	33
2.4.2 Heterogenität	35
2.4.3 Nutzung verteilter Geodatenbanken	36

3 Lösungsansatz für den direkten Datenbankzugriff	40
3.1 Theoretischer Ansatz	40
3.2 Methoden	42
4 Direkter Datenbankzugriff in heterogenen GIS-Architekturen	44
4.1 Testumgebung für direkte Datenbankzugriffe	44
4.1.1 Testarchitektur	44
4.1.2 Geodaten	49
4.2 Testverfahren	49
4.2.1 Implementierung der Geodatenverwaltung in den Datenbanksystemen	50
4.2.2 Lesende Datenbankzugriffe über heterogene GIS-Clients	55
4.2.3 Datenbankverbindungen und Programmierschnittstellen	70
5 Ergebnisse und Interpretation	78
5.1 Modelle und Technologien	78
5.2 Anforderungen und Einsatzbereiche	81
5.3 Integration in IT- und GIS-Architekturen	83
6 Zusammenfassung und Ausblick	86
6.1 Zusammenfassung	86
6.2 Ausblick	88
Literaturverzeichnis	89
Anhang	97
Räumliche Datentypen - Definition, Konstruktion und Zugriff	97

Abbildungsverzeichnis

Abb. 1: Struktur der Master Thesis	7
Abb. 2: Geodatenhaltungskonzepte	21
Abb. 3: Geometrieklassen des Simple-Feature-Modells	28
Abb. 4: SQL/MM Spatial Geometriemodell	30
Abb. 5: Beispiel einer webbasierten N-Tier-Architektur	34
Abb. 6: Gliederung der Datenbank-Heterogenität	36
Abb. 7: Möglichkeiten zur Nutzung verteilter Geodatenbanken	37
Abb. 8: Software und Architektur der Testumgebung	45
Abb. 9: Beispiel einer Linie in SDO_GEOMETRY	51
Abb. 10: Zweischichtige GIS-DB-Architektur	57
Abb. 11: Beispiele für Simple Feature Geometrien	59
Abb. 12: Beispiele für Geometrien mit Kreisbögen	61

Tabellenverzeichnis

Tab. 1: Räumliche Datentypen/Speicherformate	97
Tab. 2: Konstruktoren und Zugriffsmethoden	98

Abkürzungsverzeichnis

CORBA	Common Object Request Broker Architecture
DB	Datenbank
DBMS	Datenbankmanagementsystem
DBS	Datenbanksystem
DIN	Deutsches Institut für Normung e. V.
DHDN	Deutsche Hauptdreiecksnetz
GI	Geoinformation
GIS	Geographisches Informationssystem
GML	Geography Markup Language
ISO	International Organization for Standardization
IT	Informationstechnologie
J2EE	Java 2 Enterprise Edition
OGC	Open Geospatial Consortium
ORDBMS	Objektrelationales Datenbankmanagementsystem
RDBMS	Relationales Datenbankmanagementsystem
SFA	OpenGIS® Implementation Specification for Geographic information - Simple feature access
SOA	Serviceorientierte Architektur
SQL	Structured Query Language
SQL/MM	SQL Multimedia
WKB	Well-known Binary
WKT	Well-known Text
VDB	Verteilte Datenbanken

1 Einführung

In diesem einführenden Kapitel soll die allgemeine Problemstellung aufgezeigt und die sich daraus ableitenden, spezifischen Fragestellungen für diese Master Thesis dargelegt werden. Des Weiteren wird die Theorie und Methodik des Lösungsansatzes vorgestellt.

1.1 Problemstellung und Motivation

In vielen Institutionen sowohl aus dem öffentlichen als auch aus dem privaten Sektor hat sich der Einsatz von Geodaten und Geoinformationssystemen zur Unterstützung der Geschäftsprozesse schon seit vielen Jahren etabliert. Ein weiterer Zuwachs von GIS-gestützten Verfahren ist aufgrund der heutig stark ausgeprägten Verfügbarkeit von geographischen Daten sowie einem gesteigerten Bewusstsein für den fachlichen und wirtschaftlichen Mehrwert durch geographische Informationssysteme zukünftig zu erwarten. In Folge dessen und im Zuge einer breiten Informationsintegration und -vernetzung sowie neuer Anforderungen an die Datenerhebung und -analyse befinden sich aktuell vor allem in mittleren und großen Organisationen unternehmensweite Geoinformationssysteme und Geodateninfrastrukturen im Auf- oder Ausbau. In diesem Kontext sind sowohl die GIS-internen Verbindungen und Strukturen als auch der Zugriff auf die Geodaten von und die Koppelung mit fachlich anderen IT-Anwendungen zu betrachten.

Diese Organisationen kennzeichnen sich durch verteilte, fachlich verschiedene Einheiten mit individuellen Anforderungen innerhalb eines übergreifenden Verbundes. Dadurch liegen oftmals aus historischen, fachlichen oder strukturellen Gründen heterogene Daten und Systemlandschaften vor. Die Geodaten werden dabei häufig in Datenbanksystemen, wie andere Unter-

nehmensdaten, verwaltet. In der Entwicklung und im Ausbau von organisationsweiten GIS-Architekturen, der Geodatenintegration und -vernetzung sind diese Daten und Anwendungen effizient zusammenzuführen. Die zum Teil monolithischen und gekapselten Bereiche sollen zu einem Netzwerk zur interoperablen Datennutzung umgestaltet werden, um die breite und mehrfache Nutzung sowie den wirtschaftlichen Einsatz von Geodaten zu erhöhen. Hierzu stehen unterschiedlichste Modelle, Integrationskonzepte, Technologien und Schnittstellen zur Verfügung, deren Kombination auf die jeweils vorherrschenden Verhältnisse abzustimmen ist.

Diese Master Thesis beschäftigt sich im Rahmen dieser Thematik nun konkret mit der technischen und strategischen Eignung von direkten Zugriffsschnittstellen zu Geodatenbanksystemen für die interoperable Geodatenutzung in heterogenen GIS-Landschaften. Zur Verwaltung, der Speicherung, den Zugriff und Austausch von Geoinformationen in und aus Datenbanksystemen stehen zahlreiche räumliche Datenmodelle und -formate, Zugriffsschnittstellen und technische Konzepte zur Verfügung, die einerseits durch Standardisierungsorganisationen und -konsortien und andererseits von Softwareherstellern generiert und bereitgestellt werden. Im Mittelpunkt der wissenschaftlichen Untersuchung soll nun die technische Qualifizierung dieser Modelle und Technologien für den direkten Zugriff von unterschiedlichen Geoinformationssystemen oder vergleichbaren Applikationen auf Geodaten in Datenbanksystemen sowie der sich daraus ableitende strategische Einsatz dieses direkten Datenbankzugriffs in der Praxis stehen.

1.2 Fragestellungen und Ziele

Die konkrete Zielsetzung dieser Master Thesis besteht im Kontext der beschriebenen Thematik darin, Erkenntnisse zur technischen Qualifizierung und dem strategischen Einsatz von direkten Zugriffen bzw. Zugriffsschnitt-

stellen zu geographischen Datenbanksystemen für die interoperablen Geodatennutzung in heterogenen GIS-Architekturen zu gewinnen. Die folgenden Fragestellungen sollen zur Zielhinführung und -erreicherung dienen und im Rahmen dieser Ausarbeitung betrachtet und diskutiert werden:

- Wie lassen sich die existierenden Modelle und Technologien zur technischen Realisierung des direkten Zugriffs auf Geodaten in Datenbanksystemen einordnen und wie tragen sie zu dieser bei?
- Welche technischen Voraussetzungen müssten in heterogenen Systemlandschaften seitens der Geodatenbank- und Geoinformationssysteme erfüllt sein, um einen direkten Zugriff zur interoperablen Geodatennutzung zu ermöglichen? Welche Einschränkungen sind zu berücksichtigen?
- Für welche Anforderungen und Einsatzbereiche sind direkte Zugriffe auf Geodatenbanksysteme in heterogenen Architekturen zur interoperablen Geodatennutzung geeignet?
- Wie kann die Integration dieses Ansatzes des direkten Zugriffs in eine organisationsweite IT- und GIS-Strategie respektive -Landschaft, vornehmlich auch im Hinblick auf serviceorientierte Architekturen, aussehen?

Die Fragestellungen verdeutlichen, dass es sich bei der Zielsetzung nicht um die Beantwortung von spezifischen Einzelbeobachtungen und -ausprägungen, sondern vielmehr um eine technisch-orientierte, gesamtheitliche Betrachtung der Thematik mit einer verallgemeinernden Aussage als Ergebnis handelt. Die potenzielle Unschärfe einer solchen Analyse aufgrund der vielfältigen Modelle, Technologien, Produkte, Schnittstellen und Konzepte sowie der stetigen Entwicklungen ist dem Autor bewusst und muss für die praktische Anwendbarkeit der Ergebnisse berücksichtigt werden.

1.3 Lösungsansatz

In diesem Kapitel wird der gewählte Lösungsansatz zusammengefasst dargestellt. Die detaillierte Beschreibung erfolgt in Kapitel 3.

1.3.1 Theorie

Der theoretische Ansatz beruht auf der Annahme, dass die Verwaltung, Speicherung und Abfrage von Geodaten in einem Datenbanksystem durch Normen und Standards der Geoinformationsverarbeitung allgemein spezifiziert ist. Sofern eine Implementierung dieser Standards stattgefunden hat, ist der direkte Zugriff von verschiedenen, herstellerspezifischen Geoinformationssystemen auf die Geodaten in einem oder mehreren herstellerunterschiedlichen Datenbanksystemen grundsätzlich möglich. Aufgrund des Vorkommens sehr komplexer geometrischer oder topologischer Eigenschaften bezüglich der Abbildung von realweltlichen Objekten in Informationssystemen und eines beschränkten Spezifizierungsumfanges in Standardisierungsprogrammen sind allerdings Einschränkungen hinsichtlich der Lesbarkeit von geometrischen Objekten oder deren Modellierung zu erwarten. Des Weiteren ist davon auszugehen, dass herstellerspezifische Umsetzungen oder Erweiterungen zusätzliche Interoperabilitätsgrenzen aufstellen werden.

In Bezug auf den generellen Einsatz und die Integration in eine organisationsweite IT- und GIS-Strategie wird die Hypothese aufgestellt, dass dieses Konzept der direkten Datenbankzugriffe in bestimmten Anwendungsfällen oder für spezielle Anforderungen sowie unter gewissen Voraussetzungen innerhalb einer Organisation bzw. eines Unternehmensnetzwerkes als ein adäquates Verfahren für die interoperable Geodatennutzung und den Geodaten austausch in heterogenen (GIS-)Softwarearchitekturen angesehen werden kann.

1.3.2 Methoden

Die Methodik setzt zunächst auf einer eingehenden Recherche und Analyse der aktuell verfügbaren, sowohl standardisierten als auch proprietären Modelle, Technologien und Konzepte auf. Das weitere Vorgehen besteht in der Installation einer exemplarischen Testumgebung auf Basis verschiedener Geodatenbank- und Geoinformationssysteme, anhand derer spezifizierte Testfälle durchlaufen werden. Die technische Empirie dient zur Evaluation des theoretischen Ansatzes und zur Ableitung von verallgemeinerten Aussagen der wissenschaftlichen Untersuchung.

1.3.3 Werkzeuge

Als Hauptinstrumentarium für die praktische Untersuchung wird eine Testumgebung aus unterschiedlichen Softwareprodukten eingesetzt. Diese Architektur besteht aus nach speziellen Kriterien ausgewählten Geodatenbank- und Geoinformationssystemen. Eine detaillierte Beschreibung erfolgt in Kapitel 4.1.1.

1.3.4 Daten

Die praktische Untersuchung fokussiert sich gemäß der Zielsetzung auf den allgemeinen, technisch möglichen Zugriff bzw. die Zugriffsschnittstellen. Die Eignung oder Notwendigkeit zur Verwendung eines bestimmten Testgebietes für die Analyse besteht nicht. Vielmehr müssen hier standard- oder herstellerkonforme Musterdatensätze und -objekte generiert und für den Ablauf der Testfälle eingesetzt werden, um die Aussagekraft und Sicherheit der Ergebnisse zu gewährleisten.

1.4 Thematische Einschränkung

Die Vielzahl an standardisierten und proprietären Modellen, Technologien, Schnittstellen, Konzepten und Produkten erfordert eine Einschränkung des

Arbeitsfeldes, um die definierten Ziele zu fokussieren und thematisch klar abzugrenzen. Nur somit kann im verfügbaren Rahmen eine adäquate Bewertung und Aussage erzielt werden. Folgende Eingrenzungen gelten für diese Ausarbeitung (Nicht-Ziele):

- Eine Marktanalyse bzw. der Vergleich von allen verfügbaren Softwareprodukten hinsichtlich ihrer implementierten Technologien und Konzepten ist aufgrund des äußerst großen Angebotes weder zu leisten noch im Sinne einer wissenschaftlichen Untersuchung und wird deshalb nicht durchgeführt.
- Die Konformität der eingesetzten Softwareprodukte zu den relevanten Normen und Standards bzw. der Grad der Implementierung wird nicht untersucht. Ebenso dienen Überprüfungen der allgemeinen Funktionsfähigkeit oder -tauglichkeit der Schnittstellen, spezielle Qualitätsprüfungen sowie die Softwarekritik oder -empfehlung nicht der Zielerreichung und sind nicht Bestandteil dieser Arbeit. Evaluationen und Informationen hierzu werden über herstellereigene Tests, Konformitätstests, Testplattformen oder den praktischen Einsatz nachgewiesen und bereitgestellt.
- Die semantische Interoperabilität von Daten bzw. Datenmodellen und deren Betrachtung ist eine gänzlich andere Zielrichtung als die technische und wird nicht untersucht.
- Die Untersuchung beschränkt sich weiterhin auf den lesenden Zugriff von zweidimensionalen, vektoriellen Geodaten. Die Untersuchung beinhaltet folglich keine topologischen Modelle, Netzwerke, drei oder mehrdimensionale Modelle, lineare Referenzierungen, räumlich-zeitliche Modelle oder weitere, spezialisierte Geodatenmodelle oder Ausprägungen.

1.5 Struktur der Master Thesis

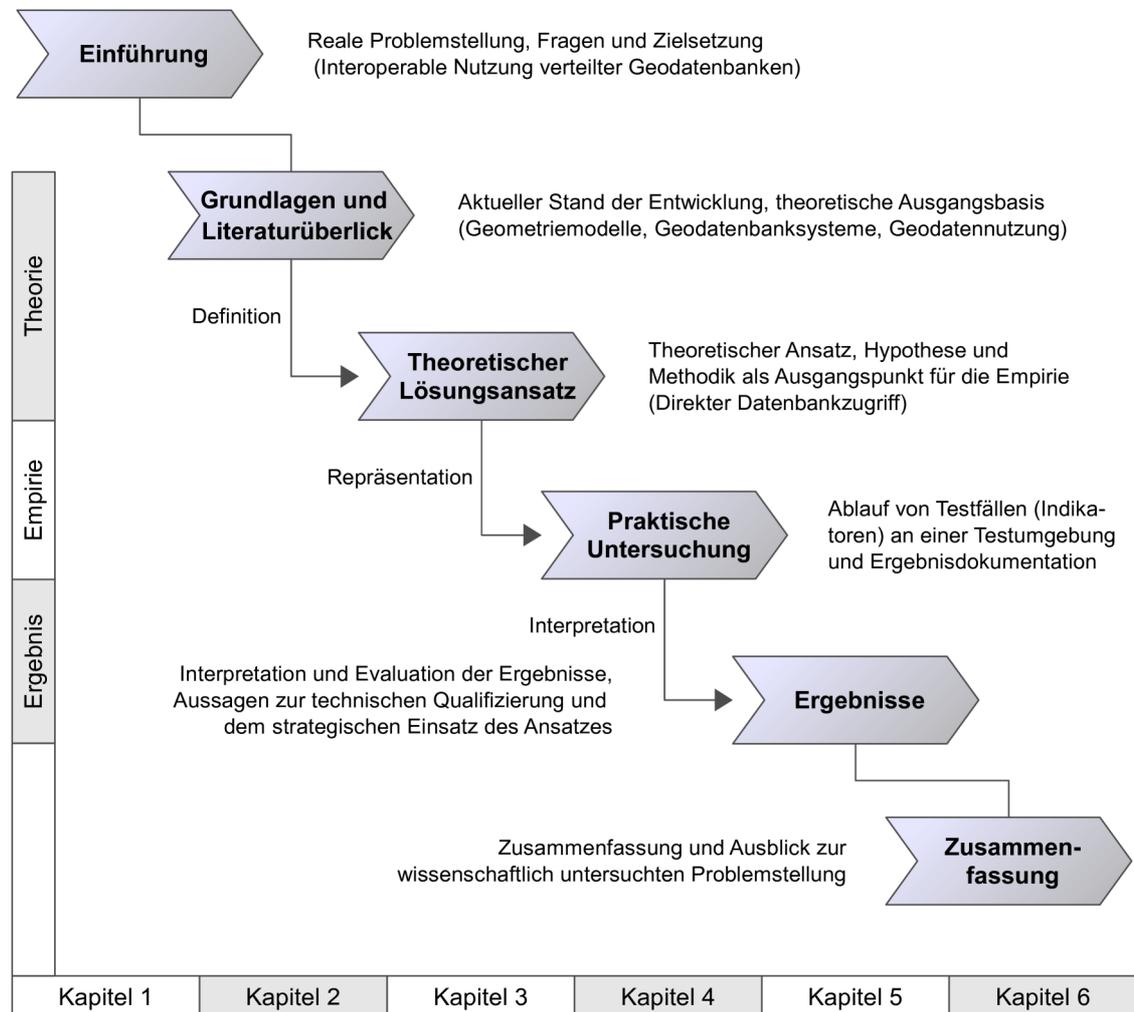


Abb. 1: Struktur der Master Thesis (Entwurf von Göttmann 2009)

Die Einführung in diese Master Thesis mit der allgemeinen Problemstellung, den konkreten Fragen und Zielen sowie dem gewählten Lösungsansatz zur Erreichung dieser wird zusammenfassend in Kapitel 1 gegeben.

Im Kapitel 2 werden zunächst die relevanten Begriffe Interoperabilität und Standardisierung aus Sicht der Geoinformatik betrachtet. Vor allem widmet sich dieser Abschnitt aber den theoretischen Grundlagen und dem aktuellen

Stand der Entwicklungen im Bereich der Geodatenbanksysteme, der räumlichen Datenmodelle und -schemata sowie der verteilten und interoperablen Geodatennutzung in heterogenen Architekturen, und bildet somit die theoretische Ausgangsbasis für diese Arbeit.

Das anschließende Kapitel 3 übernimmt die Herleitung des theoretischen Ansatzes über direkte Datenbankzugriffe und stellt eine Hypothese als Ausgangspunkt für die wissenschaftliche Untersuchung auf. In diesem Kontext wird die Definition eines direkten Datenbankzugriffs vorgenommen. Die methodische Vorgehensweise zur Evaluation dieses theoretischen Ansatzes über eine technisch-orientierte Empirie wird hier ebenso detailliert beschrieben.

Die Durchführung der praktischen Untersuchung mittels Testfällen als Indikatoren in einer Testumgebung aus heterogenen Geodatenbank- und Geoinformationssystemen und die auftretenden Ereignisse werden im vierten Kapitel umfassend dokumentiert.

Im fünften Kapitel wird die Interpretation und Evaluation der Ergebnisse vollzogen. Die Fragestellungen und Ziele dieser Master Thesis werden dabei aufgegriffen und der theoretische Ansatz mit der Empirie verglichen. Allgemeine Aussagen zur technischen Qualifizierung und dem strategischen Einsatz des Konzeptes der direkten Datenbankzugriffe beschließen dieses Kapitel als Ergebnis dieser Arbeit.

Das letzte Kapitel gibt eine Zusammenfassung dieser Ausarbeitung und einen Ausblick zur wissenschaftlich untersuchten Problemstellung.

2 Grundlagen und Literaturanalyse

Das Kapitel Grundlagen und Literaturanalyse dokumentiert die Literatur und den gegenwärtigen Stand in den für diese Arbeit relevanten Themenbereichen und liefert die Grundlagen zur Thematik.

2.1 Literaturüberblick

Die Zielsetzung dieser Master Thesis verbindet sowohl technische als auch strategische Fragestellungen zur Thematik der Nutzung verteilter, heterogener Geodatenbanken über direkte Datenbankschnittstellen und bedingt folglich die Betrachtung und Verknüpfung verschiedener Teilbereiche von Geoinformationssystemen wie Geodatenmodelle, Datenbanksysteme, Standardisierung, Interoperabilität, Schnittstellen und GIS-Architekturen. Die relevanten Quellen spiegeln diese inhaltliche Vielfalt und Überschneidung wider und dokumentieren den gegenwärtigen Stand der Entwicklungen auf diesen Gebieten der Geoinformatik.

Das Grundlagenwissen zur Modellierung, Abfrage und Analyse von Geodaten in geographische Datenbanksystemen sowie deren Architektur, Zugriffsschnittstellen und Anwendungsfelder findet sich in Fachliteratur der angewandten Geoinformationswissenschaft wie beispielsweise von Brinkhoff (2008) und Yeung & Hall (2007). Die Nutzung und Integration von Geodaten mittels Datenbanksystemen wurde in wissenschaftlichen Arbeiten u. a. von Stolze (2006), Scheugenpflug (2005) und Donaubaue (2004) untersucht. Des Weiteren sind im Kontext der Modellierung, Speicherung, Verwaltung und des Zugriffs von Geodaten in Datenbanksystemen und der Geodateninteroperabilität die Normen der International Organization for Standardization (ISO) und die Open Geospatial Consortium (OGC) Spezifikationen wie

- ISO 19107 Spatial schema
- ISO 19125-1 Simple feature access - Part 1: Common architecture bzw. OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture
- ISO 19125-2 Simple feature access - Part 2: SQL option bzw. OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option
- ISO 19136 Geography Markup Language (GML) bzw. OpenGIS® Geography Markup Language (GML) Encoding Standard
- ISO/IEC 13249-3 Information technology - Database languages - SQL multimedia and application packages - Part 3: Spatial

von grundlegender Bedeutung.

Neben den fachspezifischen Quellen des Geoinformationssektors ist Literatur der allgemeinen Informationstechnologie zu Datenbanksystemen und -schnittstellen (u. a. Saake et al. 2008), zur Informationsintegration (u. a. Leser & Naumann 2007) und zu Systemarchitekturen (u. a. Dunkel et al. 2008) heranzuziehen, da sich hier die Umsetzungen im Geoinformationsbereich sehr stark an die allgemeinen IT-Standardisierungen und -Konzepte anlehnen.

Nicht zuletzt müssen die bereits angesprochenen, aktuellen Entwicklungen seitens der IT-Branche respektive die Softwareprodukte berücksichtigt werden, da hier die wissenschaftlichen Erkenntnisse, die herstellereigenen Forschungen und die Standardisierungen der relevanten Organisationen und -gremien unmittelbar einfließen. Hierzu können Dokumentationen oder Whitepapers der Softwarehersteller ausgewertet werden. Im Rahmen dieser

Master Thesis wurden diese für die Durchführung des technischen Teils der Empirie verwendet.

Die in diesem Kapitel vorgestellten Informationsquellen bilden die theoretische Grundlage dieser Arbeit und vermitteln den aktuellen Stand der Entwicklung und der praktischen Anwendung in den relevanten Themengebieten. Dieser wird in den folgenden Kapiteln zusammengefasst und grundlegend dargestellt.

2.2 Interoperabilität und Standardisierung

In diesem Kapitel soll auf die im Kontext dieser Arbeit wichtigen Begriffe Interoperabilität und Standardisierung aus Sicht der Geoinformatik eingegangen werden. Die Intention ist hierbei nicht der Anspruch auf eine umfassende Darstellung, sondern vielmehr soll mit dieser kurzen Einführung aufgezeigt werden, dass diese für die systemübergreifende Geodatennutzung und -integration bedeutenden Begriffe bereits mit diversen Definitionen belegt sind. Dies verdeutlicht die Notwendigkeit zur Auseinandersetzung mit diesen als Einstieg in die Thematik und zur Festlegung der Begriffsverwendungen im Rahmen dieser Ausarbeitung.

2.2.1 Interoperabilität

Der Begriff Interoperabilität ist im Allgemeinen nicht auf eine bestimmte Domäne beschränkt und folglich existieren unterschiedlichste Definitionen in Abhängigkeit des fachlichen Kontextes¹. Im Folgenden wird nur die im Rahmen dieser Arbeit relevante Bedeutung für die Geoinformatik näher betrachtet. In den Publikationen finden sich selbst in diesem speziellen Wissensgebiet verschiedene Definitionen und Erläuterungen.

¹ Eine Internetrecherche bestätigte die weitverbreitete Verwendung wie z. B. in der Informatik, Telekommunikation, im Bahnverkehr, in der Industrie sowie beim Militär.

Das Open Geospatial Consortium weist in seinem Whitepaper zur „Interoperabilität und offene Architekturen“ (OGC 2005a) auf diese Begriffsmehrdeutigkeit in der Geoinformationsbranche hin und definiert Interoperabilität als einen umfassenden, fort dauernden Prozess zur Sicherstellung eines optimalen Austauschs und der Nutzung von Informationen und Diensten innerhalb und zwischen Organisationen, der dabei sowohl die technischen, semantischen, organisatorischen, strategischen und rechtlichen Bereiche als notwendige Bestandteile zur Umsetzung mit einbezieht. Das Konsortium selbst versucht vor allem die Interoperabilität auf der technischen Ebene, d. h. die Kombinierbarkeit, das Zusammenspiel und die Kommunikation von unterschiedlichen Softwaresystemen, durch die Erarbeitung und Publikation von geeigneten Standards voranzubringen (OGC 2004). Im Kontext dieser OGC Spezifikationen wird Interoperabilität von der OGC (2009, OGC Glossar) wie folgt definiert: *“Interoperability, (...), is software components operating reciprocally (working with each other) to overcome tedious batch conversion tasks, import/export obstacles, and distributed resource access barriers imposed by heterogeneous processing environments and heterogeneous data.”*

Für gesetzgebende Institutionen steht Interoperabilität für die Kombinierbarkeit von Geodaten und die Interaktion von Diensten bzw. Systemen unter Berücksichtigung von Standards (vgl. INSPIRE 2007, Kap. 1, Art. 3 (7) und GeoZG 2009, §3 (4)). Die rechtlichen Bestimmungen beschreiben Interoperabilität auf einer mehr allgemeinen, ganzheitlichen Ebene und gehen nicht auf technische Detailspezifikationen ein. Sie geben aber unmissverständlich die Durchführung auf Basis von Standards vor. Die konkrete Umsetzung wird in anhängenden Durchführungsbestimmungen geregelt.

Zur Hinführung auf den Fokus dieser Arbeit wird Interoperabilität nachfolgend nochmals aus einer mehr technischen Sichtweise betrachtet, unter der

diese allgemein als die Fähigkeit von zwei oder mehr Systemen zur harmonischen Zusammenarbeit beschrieben werden kann (Yeung & Hall 2007). Im Kontext von Datenbanken spezifizieren Yeung & Hall (2007) näher, dass sie sich hierbei auf den Austausch und die Integration von Daten unterschiedlicher Datenbanken unabhängig von Format und Ursprung bezieht. Donaubaue (2004) definiert in seiner wissenschaftlichen Arbeit zur interoperablen Nutzung verteilter Geodatenbanken mittels standardisierter Geo Web Services Interoperabilität als die Fähigkeit a priori autonomer Systeme zur Zusammenarbeit über eindeutig spezifizierte Schnittstellen, indem sie gegenseitig Dienstleistungen in Anspruch nehmen.

Aufgrund der bereits genannten und auszugsweise dargelegten Diversität der Begriffsbestimmung wird im Rahmen dieser Arbeit, unter Berücksichtigung der vorausgehenden Darstellungen unterschiedlicher Sichtweisen, Folgendes unter einer „interoperablen Geodatennutzung“ verstanden:

Interoperable Geodatennutzung bezeichnet die gegenseitige Nutzung von Geodaten über definierte Schnittstellen in heterogenen Systemlandschaften. Die Nutzung der Daten kann dabei funktional verschieden sein und sowohl den Zugriff, die Manipulation, die Analyse oder andere Ausprägungen umfassen.

2.2.2 Standardisierung

Die effektive und harmonische Zusammenarbeit von unterschiedlichen Systemen, die vor allem den Austausch und die Nutzung von Daten und Funktionalitäten impliziert, ist in der Informationstechnologie (IT) schon seit langem durch die Verwendung von gemeinsamen Regeln geprägt. Im Kontext von Regelwerken existieren weltweit nicht nur zahlreiche Begriffsdefinitionen, sondern auch unterschiedliche Standardisierungsinstitutionen sowie Ausprägungen, Gültigkeitsansätze und Detailgrade von Standards

(Yeung & Hall 2007), deren eindeutige Bestimmung und Erläuterung aber weder Absicht dieser Arbeit ist und noch in deren Rahmen umfassend vorgenommen werden kann. An dieser Stelle sollen aber zumindest die für die Geoinformatik bedeutenden Standardisierungsinstitutionen genannt und eine vereinfachte Klassifizierung von Standards gegeben werden.

In der Geoinformationswelt haben in den letzten Jahren vor allem zwei Institutionen, die sich mit der Entwicklung und Publikation von technischen Standards beschäftigen, eine besondere Bedeutung erlangt:

- Die International Organization for Standardization (ISO) ist die weltweit bedeutendste Standardisierungsorganisation, die aktuell 157 nationale Standardisierungsinstitutionen repräsentiert und zu einem Netzwerk vereint. Ein technisches Komitee „Geographic information/Geomatics“ (ISO/TC211) entwickelt speziell Standards im Bereich der digitalen geographischen Informationen und hat bisher 42 Standards für die Geoinformationsbranche veröffentlicht (ISO 2008).
- Das Open Geospatial Consortium, Inc (OGC) ist ein internationales Industriekonsortium mit aktuell 367 Mitgliedern aus Wirtschaft, Verwaltung und Wissenschaft, das in gemeinschaftlichen Entwicklungsprozessen öffentlich zugängliche Spezifikationen für die Geoinformationsverarbeitung erstellt (OGC 2009). Das ISO/TC211 und das OGC arbeiten bei ihren Standardisierungsbestrebungen eng über eine gemeinsame Koordinierungsgruppe zusammen (OGC 2005a).

Eine vereinfachte Klassifikation von Standards kann wie folgt vorgenommen werden:

Der Begriff „Standard“ als Regelwerk oder Vorschrift wird in verschiedenen Konstellationen und in der Literatur nicht immer eindeutig verwendet, und es kommt mitunter zu einer Begriffsverwirrung. In deutschsprachigen Pu-

blikationen zur Geoinformatik findet sich teilweise eine klare, inhaltliche Begriffstrennung zwischen Norm als de-jure-Standard und Standard im Sinne von de-facto-Standard. Das Wort Standard ist aber auch als allgemeine Bezeichnung, unabhängig davon, welche Art von Regelwerk gemeint ist, sehr weit verbreitet. Des Weiteren wird das Wort in Kombination als Industrie-, herstellerepezifischer oder offener Standard verwendet. Im englischen Sprachgebrauch hingegen kommt das Wort „Standard“ als Begriff für alle Arten von Regelwerken (auch für das deutsche Wort Norm), Verordnungen, Spezifikationen etc. vor und wird ggf. durch weitere Wortergänzungen näher beschrieben.

Der Begriff „Norm“ (de-jure-Standard) bezeichnet allgemein gültige, rechtlich anerkannte Regeln, Vorschriften oder Leitlinien, die ein Normungsverfahren durchlaufen haben und durch eine anerkannte (inter)nationale Standardisierungsorganisation (z. B. ISO, Deutsches Institut für Normung e. V. (DIN)) angenommen wurden (Datacom 2008).

Ein „de-facto-Standard“ (Industriestandard) ist ein technisches Regelwerk, das sich über längere Zeit in der Praxis bewährt und eine weite Verbreitung und Akzeptanz gefunden hat. Ein de-facto-Standard wird nicht durch ein (inter)nationales Normungsverfahren, sondern meist nur durch eine Organisation, erstellt und ist nicht offiziell anerkannt, kann ggf. aber später als Norm aufgenommen werden (Datacom 2008). In diesem Zusammenhang sind auch die zunächst als de-facto-Standards anzusehenden Spezifikationen der OGC zu nennen, die durch die zuvor beschriebene Kooperation mit der ISO häufig den Eingang in den offiziellen Normungsprozess finden und zu de-jure-Standards werden (OGC 2005a).

Ein „offener Standard“ ist dokumentiert, öffentlich zugänglich und kann kostenfrei oder kostenpflichtig (Schutzgebühr) bezogen werden. Die Teil-

und Einflussnahme am Entwicklungsprozess steht interessierten Parteien vorbehaltlos offen. Die Urheberrechte und Patente verbleiben beim Herausgeber. Dritten ist es uneingeschränkt gestattet, Produkte gemäß diesem Regelwerk zu erstellen bzw. diese zu implementieren. Normen wie z. B. von der ISO oder Spezifikationen der OGC können weitestgehend als offene Standards betrachtet werden (Yeung & Hall 2007).

Ein „herstellerspezifischer (proprietärer) Standard“ wird durch ein Unternehmen entwickelt, das dementsprechend alleinig für die Entwicklung und Pflege zuständig ist und alle Rechte und Patente besitzt. Dritten wird es ggf. gestattet diesen Standard zu nutzen. Ein Beispiel für einen weitläufig genutzten, proprietären Standard, der sich in der Geoinformationsverarbeitung mittlerweile als de-facto-Standard etabliert hat, ist das ESRI Shapefile-Format (Yeung & Hall 2007).

Um der allgemein vielfältigen Begriffsverwendung entgegenzuwirken, wird im Rahmen dieser Arbeit das Wort „Standard“ als Überbegriff für alle Arten von Regelwerken eingesetzt. Eine genaue Unterscheidung wird dann vorgenommen, wenn es inhaltlich bedeutend ist, die spezielle Ausprägung kenntlich zu machen.

2.3 Geodatenbanksysteme und Geodaten

2.3.1 Geodatenbanksysteme

Die heutigen Geoinformationssysteme werden im Allgemeinen durch vier Komponenten für die Aufgaben Erfassung, Verwaltung (Datenmodellierung und Speicherung), Analyse und Präsentation charakterisiert. Die Verwaltung und Speicherung der Geodaten erfolgt dabei heutzutage zunehmend in Datenbanksystemen (Brinkhoff 2008).

Datenbanksysteme

Datenbanksysteme (DBS) moderner Technologie dienen zur effizienten Verwaltung großer Datenmengen und definieren ein Datenbankmodell zur einheitlichen, formalen Beschreibung von Daten und ihrer Beziehungen untereinander. Sie bieten Sprachen und Operationen zur Definition, Anfrage und Manipulation der Daten und interne Optimierungsstrukturen für den schnellen Zugriff. Dabei können durch entsprechende Transaktionskonzepte viele Benutzer gleichzeitig auf gemeinsame Datenbestände zugreifen. Des Weiteren stehen umfangreiche Sicherungsmethoden zur Verfügung, die vor allem den Datenschutz, die Datenintegrität (Konsistenz) und die dauerhafte Sicherheit gegen Datenverlust gewährleisten (Saake et al. 2008).

Heute sind relationale Datenbankmanagementsysteme (RDBMS) auf Basis des relationalen Datenbankmodells, das 1970 erstmals durch Edgar F. Codd beschrieben wurde, mit rund 80% Marktanteil am weitesten verbreitet (Brinkhoff 2006). Der Zugang zu diesen Systemen und den Daten erfolgt dabei weitestgehend über die Structured Query Language (SQL) als Datenbanksprache, die von der ISO standardisiert ist und heute von allen relationalen Datenbanksystemen unterstützt wird. Das Konzept der Objektorientierung, das in den 1990er Jahren durch objektorientierte Programmiersprachen sehr starken Einzug in die Softwareentwicklung hielt, übertrug sich auch in die Datenbankwelt. Die rein objektorientierten Datenbankmanagementsysteme (ODBMS), die erstmals Ende der 1980er Jahre auf den Markt kamen, nehmen bis heute allerdings nur eine bescheidene Rolle ein, da einige Vorteile des relationalen Modells mit der Objektorientierung nur schwer vereinbar sind. Daneben wurden aber auch verschiedene, dem objektorientierten Konzept folgende Erweiterungen in RDBMS implementiert, die allgemein unter dem Begriff „objektrelational“ zusammengefasst werden. Diese objektrelationalen Datenbankmanagementsysteme (ORDBMS) spielen insbesondere für die Verwaltung von räumlichen Daten eine wichtige Rolle (Saake et al. 2008).

Die Grundlagen zu Datenbanksystemen, (objekt-)relationalen Modellen und SQL finden sich in zahlreichen Lehrbüchern und sollen an dieser Stelle nicht weiter dargelegt werden.

Geodatenbanksysteme

„Datenbanksysteme, die die Speicherung von Geodaten und die Bearbeitung räumlicher Anfragen in hinreichender Weise unterstützen, werden räumliche Datenbanksysteme oder auch Geodatenbanksysteme (engl. Spatial Database Systems) genannt“ (Brinkhoff 2008, S. 24).

Nach der Definition von Güting (1994) wird ein Geodatenbanksystem durch drei wesentliche Eigenschaften charakterisiert:

- Es ist ein vollständiges Datenbanksystem mit zusätzlichen Möglichkeiten zur Verwaltung von räumlichen Daten.
- Es bietet räumliche Datentypen in seiner Datenmodellierungs- und Abfragesprache.
- Es implementiert zumindest räumliche Indizierung und effiziente Algorithmen für räumliche Beziehungen.

Ein Geodatenbanksystem (GDBS) ist den vorausgehenden Definitionen nach keine spezielle, individuelle Software, sondern ein herkömmliches Datenbanksystem, das über entsprechende Funktionalitäten zur Speicherung und Verwaltung von raumbezogenen Daten verfügt.

Anforderungen an Geodatenbanksysteme

Brinkhoff (2008) stellt als Anforderungen an ein GDBS, vergleichbar mit Güting (1994), die Bereitstellung von geometrischen Datentypen, die die Geometrie eines Geoobjektes (z. B. Punkte, Linien, Flächen) beschreiben, von Methoden für diese zur Ausführung von geometrischen Funktionen (z. B. Längen- oder Schnittberechnung) und von optimierten und effizienten Verarbeitungsabläufen räumlicher Basisanfragen und anderer geometrischer

Operationen durch z. B. räumliche Indexe und Algorithmen für die Beantwortung räumlicher Fragestellungen. Des Weiteren muss das Datenmodell zur Integration der Geodaten in die bestehende IT-Infrastruktur einer Organisation und deren Geschäftsprozesse allgemein anerkannte Standards einhalten.

Yeung & Hall (2007) spezifizieren als zusätzliche Charakteristika, die speziell für Geodatenbanksysteme relevant sind, die Sicherstellung von Integrität vor allem hinsichtlich Topologie und Semantik über entsprechende Bedingungen und die Gewährleistung von langen Transaktionen, die gerade bei der Geodatenverarbeitung häufig auftreten können, über adäquate Datenbankmanagementkonzepte.

Vorteile der datenbankbasierten Datenhaltung

Die Entwicklung hin zur Ablage von Geodaten in Datenbanksystemen setzte in der GIS-Welt etwa Mitte der 1990er Jahre durch steigende Anforderung an die GIS-Datenhaltung und die Bewegung hin zu offenen Geoinformationssystemen zur nahtlosen Integration in die allgemeinen IT-Infrastrukturen ein (Brinkhoff 2008). Bis Mitte der 1990er Jahre wurden die Daten eines GIS nahezu ausschließlich in Dateien oder GIS-spezifischen Datenhaltungskomponenten abgelegt. Der Bedarf große Datenmengen effektiv und mit den Vorteilen bewährter Techniken von Datenbankmanagementsystemen zu verwalten, führte Ende der 1990er Jahre zur Verwendung von relationalen DBMS (Yeung & Hall 2007). Die später aufkommenden objektrelationalen Datenbanksysteme mit der Fähigkeit der objektrelationale Modellierung eigneten sich für die Speicherung und Verwaltung von Geodaten aus verschiedener Hinsicht besser als die relationalen Strukturen (Scheugenpflug 2005), weshalb sie sich heute weithin durchgesetzt haben (siehe Abbildung 2).

Die folgende Zusammenfassung zeigt die wesentlichen Vorteile die Datenbanksysteme zur Verwaltung und Speicherung von Geodaten gegenüber der

Ablage in Dateien bieten (Brinkhoff 2008, Greener 2008):

- Zentrale Datenhaltung und zeitgleicher Mehrbenutzerbetrieb (Konsistenz durch Transaktionskonzepte)
- Datenunabhängigkeit (Anwendungen bleiben weitestgehend von logischen und physikalischen Änderungen der Datenhaltung unberührt)
- Datensicherungs-, Replikations- und Wiederherstellungsmechanismen durch das Datenbanksystem
- Gewährleistung der Datenintegrität
- Abfragemechanismus (Datenbanksprache)
- Fein strukturierte Zugriffskontrolle durch Benutzerverwaltung und individuelle Sichten auf die Daten
- Integration der Geoinformationen mit anderen Organisationsdaten bereits auf Datenbankebene

Trotz aller Vorteile muss im Einzelfall analysiert werden, welche Datenhaltung (Datei- oder Datenbanksystem) sich für das jeweilige Geoinformationssystem am besten eignet. Für mittlere und große Organisationen sind Datenbanksysteme heute meist die erste Alternative, wohingegen gerade bei kleinen Unternehmen oder einfachen Anwendungen eine dateibasierte Datenablage ausreichend und nicht zuletzt aufgrund des größeren Aufwands einer Datenbankarchitektur hinsichtlich Kosten und Know-how wirtschaftlicher sein kann.

Speicherung von Geodaten

Die Speicherung von Geodaten im Vektormodell kann in relationalen oder objektrelationalen Datenbanksystemen erfolgen. Zur weiteren Vertiefung wird hier auf die Lehrbücher von Brinkhoff (2008), Yeung & Hall (2007) und die wissenschaftliche Arbeit von Scheugenpflug (2005), der die beiden Mo-

dellierungs- und Datenbankkonzepte vergleicht, verwiesen. An dieser Stelle sollen die technischen Datenhaltungskonzepte zusammengefasst und in vereinfachter Form dargelegt werden.

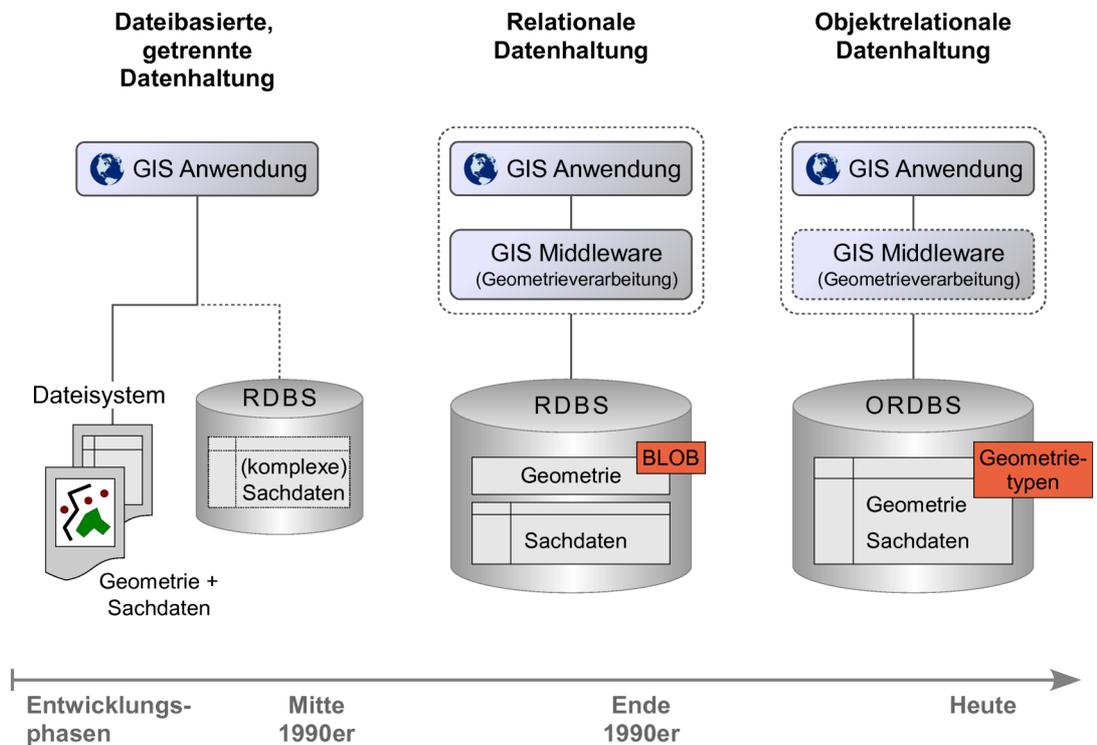


Abb. 2: Geodatenhaltungskonzepte (Entwurf von Göttmann 2009)

Bei dem klassischen Ansatz der dateibasierten, getrennten Datenhaltung werden die Geometrien mit ihren Attributen in GIS-spezifischen, proprietären Datenformaten in Dateien abgelegt. Lediglich komplexe Sachdatenmodelle der Geoobjekte werden in einem relationalen Datenbanksystem (RDBS) verwaltet und über eindeutige Schlüssel mit den Geometrien verbunden. Ebenso existieren Lösungen mit einer kompletten Trennung zwischen Geometrie- und Sachdaten, bei der alle alphanumerischen (Sach-)Attribute in einem DBS gespeichert werden. Bei dieser Form der Datenhaltung gehen vor allem die Vorteile der rein datenbankbasierten Konzepte verloren, weshalb das Gros der modernen Geoinformationssysteme daneben

ganzheitliche Datenbanklösungen anbietet.

Bei dieser ausschließlichen Speicherung und Verwaltung von Geodaten in Datenbanksystemen sind zunächst die relationalen Datenhaltungskonzepte zu nennen. Bei diesen werden alle Geometrie- und Sachdaten in einem relationalen, herstellereigenen Datenbankmodell abgelegt. Ein verbreiteter Ansatz besteht darin, die Geometrie in einem proprietären Format in einem binären Datentyp (Binary Large Object²) zu speichern. Diese binäre Speicherung verbirgt allerdings die Geometrie bzw. das Geometriemodell nicht nur vor dem GIS-Anwender und dem Datenbankentwickler, sondern ist auch für andere, als das erzeugende GIS bzw. dessen Middleware nicht direkt interpretierbar. Dies widerspricht der Idee eines offenen Systems und erhöht die Abhängigkeit zum GIS-Hersteller. Das datenhaltende DBS kann die Semantik der Binärdaten nicht erkennen und folglich keine Mechanismen zur Sicherstellung der Konsistenz, zur effizienten Anfragebearbeitung oder für geometrische Operationen bereitstellen. Daneben können die Geometrien in einem relationalen Modell auch auf mehrere Tabellen aufgeteilt und in numerischen Datentypen gespeichert werden. Hierbei werden die Zugriffstransparenz und Flexibilität bzgl. des Datenmodells gewährleistet. Die relationale Zerlegung der Geometrie wirkt sich allerdings auf die Abfragefunktionalität und -performance hinreichend negativ aus (Brinkhoff 2008, Greener 2008, Scheugenpflug 2005).

Der moderne Datenhaltungsansatz nutzt die Möglichkeiten objektrelationaler Datenbanksysteme und modelliert die Geometrien in einem räumlichen Datentyp. Seit der Einführung objektrelationaler Konzepte in Datenbanksystemen mit SQL3 (ISO-Normen SQL-99 und SQL-2003) ist es möglich nutzerdefinierte Datentypen (User-defined Data Types³ (UDT)), denen spezifische Methoden zugeordnet werden können, zu erstellen. Punkte, Linien

2 Der Datentyp Binary Large Object (BLOB) ist Bestandteil der ISO-Norm SQL-2003 und ermöglicht es, beliebige Binärdaten (z. B. Rasterbilder oder andere Multimediadaten) in einem Datenbanksystem zu speichern.

3 SQL-spezifischer Begriff für die Definition abstrakter Datentypen (Saake et al. 2008, 551 ff.)

oder Flächen werden als UDTs abgebildet und mit entsprechenden, geometrischen Methoden zur Manipulation, zu topologischen Prüfungen oder räumlichen Analysen versehen (Saake et al. 2008). Dieses Konzept ermöglicht somit die Speicherung der geometrischen Attribute in Verbindung mit den Sachattributen in einer Relation. Des Weiteren bieten die räumlichen Datentypen mit ihren Methoden umfangreiche, geometrische Funktionalitäten auf Seiten des Datenbanksystems. Da die Modellierung und Implementierung eines solch komplexen, objektrelationalen Designs mit UDTs und Methoden für den Endanwender kaum zumutbar ist, liefern einige große Datenbankhersteller bereits entsprechende Erweiterungen zur Speicherung und Verwaltung von Geodaten mit ihrer Datenbanksoftware aus. Diese können dann allerdings herstellerspezifisch aufgebaut sein und der GIS-Anwender begibt sich mit seiner Datenhaltung wieder in die Abhängigkeit eines Anbieters. Folglich müssen alle Geoinformationssysteme, die die Geodaten beziehen wollen, das individuelle Geometrieschema lesen können. Dieser Problematik kann nur über die Einhaltung von normierten Geometrie-modellen (siehe Kapitel 2.3.2) entgegengewirkt werden (Brinkhoff 2008).

Softwareprodukte

Die Palette an Geodatenbanksystemen ist im Vergleich zu anderen Softwareparten gegenwärtig noch überschaubar. Es befinden sich aktuell vier kommerzielle Datenbanksysteme, die über entsprechende Erweiterungen zur Verwaltung von Geodaten verfügen, auf dem Markt. Dazu gehören: IBM Informix (Spatial Data Blade), IBM DB2 (Spatial Extender), Microsoft SQL Server (Unterstützung von räumlichen Datentypen und Methoden ab SQL Server 2008) und Oracle (Spatial). Daneben bieten auch zwei Vertreter der Open-Source-Datenbanksysteme räumliche Funktionalitäten an. Hier ist vor allem das objektrelationale Datenbanksystem PostgreSQL mit der Erweiterung PostGIS zu nennen. Das relationale Datenbanksystem MySQL stellt bisher nur einfache räumliche Funktionalitäten zur Verfügung (Brink-

hoff 2008, Microsoft 2008). Nähere Informationen und Vergleiche zu Open-Source-Geodatenbanksysteme finden sich u. a. bei Brinkhoff (2007b) und Schanz (2006).

Datenbankverbindungen und -zugriffsschnittstellen

Die Abfrage von Daten aus einem Datenbanksystem via SQL kann auf einfache Weise über einen herstellereigenen DB-Client bzw. -Schnittstelle (häufig als Kommandozeilen-Werkzeuge verfügbar, z. B. sqlplus von Oracle oder psql von PostgreSQL) erfolgen (Yeung & Hall 2007). Um den direkten Zugriff von Anwendungsprogrammen (z. B. GIS-Software) auf die Daten eines Datenbanksystems grundsätzlich zu ermöglichen, muss neben einer bestehenden Netzwerkverbindung über ein Netzwerkprotokoll (z. B. TCP/IP) auch eine Datenbankzugriffsschnittstelle, die den Zugang zur Datenbank über eine konventionelle Programmiersprache anbietet, durch das DBMS zur Verfügung gestellt werden. Diese Programmierschnittstellen (engl. Application Programming Interface, API) offerieren Prozeduren bzw. Funktionen zur Definition, Manipulation und Abfrage von Daten. Die Abfolge eines Datenzugriffs gliedert sich dabei vereinfacht in folgende Schritte:

- Aufbau der Datenbankverbindung und Start einer Transaktion
- Ausführung von SQL-Anweisungen und Verarbeitung der Ergebnisse
- Abschließen der Transaktion und Beenden der Datenbankverbindung

Neben systemspezifischen Zugriffsschnittstellen gibt es auch hier Standardisierungen wie den SQL/CLI-Standard (ISO/IEC 9075-3:2008, Information technology - Database languages - SQL - Part 3: Call-Level Interface (SQL/CLI) (ISO 2008)), der durch die ISO genormt ist und von dem Konsortium Open-Group weiterentwickelt wird. Eine SQL/CLI-konforme Schnittstelle, die über die Call-Level Interface Konzepte hinaus auch den Zugriff auf verschiedene Datenbanksysteme über einen Treibermanager ermög-

licht, wurde von Microsoft als die „Open Database Connectivity“ (ODBC) Schnittstelle implementiert. Die „Java Database Connectivity“ (JDBC) Schnittstelle ist die standardisierte Datenbankschnittstelle für den Zugriff auf SQL-Datenbanken mit der Programmiersprache Java. Des Weiteren existieren vergleichbare Call Level Schnittstellen für sehr viele Skriptsprachen wie beispielsweise PHP oder Perl. (Saake et al. 2008).

Bei der Verwendung dieser generischen Programmierschnittstellen werden SQL-Anweisungen meist in Form von Zeichenketten als Parameter an die Prozeduren des APIs übergeben. Da bei dieser einfachen Kopplung von Programmiersprache und Datenbank über Call Level Interfaces die syntaktische oder semantische Überprüfung des SQLs erst zur Laufzeit durchgeführt wird, ist als erweiterter Ansatz die Einbettung von SQL in die Programmiersprache (Embedded SQL) möglich. Dabei werden die in den Quellcode eingestreuten SQL-Anweisungen zunächst durch einen Vorübersetzer (Precompiler) validiert und danach durch Prozeduraufrufe der Datenbankschnittstelle substituiert. Als ein Vertreter dieser Technik kann hier beispielsweise SQLJ für die Java-Plattform genannt werden (Brinkhoff 2008).

Darüber hinaus gibt es alternative Konzepte für objektorientierte Programmiersprachen wie Java oder C#, die das SQL durch eine Zwischenschicht weitestgehend verbergen und automatisch eine Abbildung von Tupeln bzw. Relationen in die Objekte der Programmiersprache vornehmen (objektrelationales Mapping). Das Java-basierte Framework Hibernate übernimmt beispielsweise ein solches Mapping (Saake et al. 2008). Hibernate Spatial ist eine spezielle Erweiterung von Hibernate für geographische Daten und unterstützt die meisten Funktionen der OGC Simple Feature Access Spezifikation (Hibernate Spatial 2008).

2.3.2 Räumliche Datenmodelle und Datenbankschemata

Die Phänomene der realen Welt werden in Geoinformationssystemen über verschiedene Eigenschaften abgebildet. Die Lage und Ausdehnung des Ob-

jektes (Geometrie) im Raum kann mittels Koordinaten eines Raumbezugs-systems beschrieben werden. Die thematischen Informationen zu einem Objekt werden als Sachdaten mit der Geometrie verknüpft. Neben dem diskreten Raumbezug können relative Beziehungen des Objektes zu anderen realweltlichen Erscheinungen auftreten, die als topologische Eigenschaften modelliert werden. Die gegenwärtige Forschung beschäftigt sich des Weiteren mit der Zuordnung von Zeiträumen und der räumlich-zeitlichen Dynamik von Geoobjekten. Zur Modellierung und Abfrage von Geoobjekten in Geoinformations- und Geodatenbanksystemen existieren standardisierte, räumliche Datenmodelle und -schemata sowie Zugriffsschnittstellen.

Spatial Schema

Die ISO-Norm 19107 Geographic information - Spatial schema (ISO 19107: 2003) definiert ein konzeptuelles Datenmodell zur Beschreibung von Eigenschaften räumlicher Objekte und spezifiziert räumliche Operationen für Zugriff, Abfrage, Verwaltung, Verarbeitung und Austausch von Objekten dieses Schemas. Sie behandelt Vektorgeometrien und Topologien bis zur dritten Dimension. Rasterdaten werden nicht berücksichtigt. Die Spezifikation untergliedert sich in zwei Hauptpakete: „Geometry“ für die geometrischen und „Topology“ für die topologischen Eigenschaften.

Es können null- (Punkte), ein- (Linien), zwei- (Flächen) und dreidimensionale (Körper) Objekte gebildet werden. Das Modell ermöglicht sehr detaillierte und ausgeprägte Geometriebeschreibung wie beispielsweise Linien aus Kreisbögen (GM_Arc) oder Bézier-Kurven (GM_Bezier), Flächen als Polygone mit Löchern oder als Dreiecke (GM_Triangle, u. a. für Dreiecksvermischung (GM_Tin)) sowie Körperoberflächen wie Zylinder (GM_Cylinder) oder Kugeln (GM_Sphere) (Brinkhoff 2008). Das Spatial Schema 19107 bietet ein sehr umfassendes, räumliches Objektmodell und bildet somit die konzeptuelle Grundlage für andere Standards und konkrete Implementierungen im Geoinformationsbereich.

Simple Feature Access

Die OGC Implementierungsspezifikation „Geographic information - Simple feature access“ (SFA) beinhaltet ein Modell für einfache, geometrische Objekte (simple feature geometry object model) und stellt „das für Geodatenbanksysteme wohl wichtigste Datenmodell“ (Brinkhoff 2008, S. 73) dar. Sie gliedert sich in einen Teil 1 (SFA - Part 1: Common architecture), der ein Objektmodell für einfache Vektorgeometrien (keine Rasterdaten) und räumliche Operationen für die Abfrage und Analyse der Objekteigenschaften allgemein und plattformunabhängig beschreibt, und einen Teil 2 (SFA - Part 2: SQL option), der ein SQL Datenbankschema für die Speicherung, Abfrage und Bearbeitung von Objekten über eine SQL Call-Level Schnittstelle definiert. Der Standard wurde durch die OGC entwickelt und erstmals im Jahre 1999 veröffentlicht. Die ISO hat im Jahre 2004 die beiden Teile als die Normen ISO 19125-1 und ISO 19125-2 übernommen. Die OGC SFA Dokumente in der Version 1.1.0 von 2005 entsprechen diesen ISO-Normen.

Das Simple-Feature-Modell implementiert ein Profil von ISO 19107:2003, das nur eine Untermenge des umfassenden, abstrakten Spatial Schemas darstellt und auch einige Unterschiede zu diesem aufweist. Die Wurzel des hierarchischen Modells bildet die abstrakte, nicht instanziierbare Klasse Geometry. Sie liefert die Eigenschaften und Methoden für alle Geometrien. Jeder Geometrie wird ein räumliches Referenzsystem (Spatial Reference System (SRS)) zugewiesen. Die Klasse Geometry hat vier Unterklassen: Punkt (Point), Linie (Curve), Fläche (Surface) und Geometriesammlung (GeometryCollection). Eine GeometryCollection kann sich aus heterogenen oder im Speziellen nur aus homogenen (MultiPoint, MultiCurve, MultiSurface) Einzelgeometrien zusammensetzen.

Die Unterklassen von Geometry sind beschränkt auf null-, ein- und zweidimensionale Geometrieobjekte (= Punkte, Linien und Flächen), die nur in der zweiten räumlichen Dimension (X, Y) vorkommen dürfen. Eine weitere wesentliche Einschränkung ist dadurch gegeben, dass die Verbindung von

Stützpunkten nur linear erfolgen darf, d. h. eine Linie darf sich beispielsweise nicht aus Kreisbögen oder Kurven zusammensetzen.

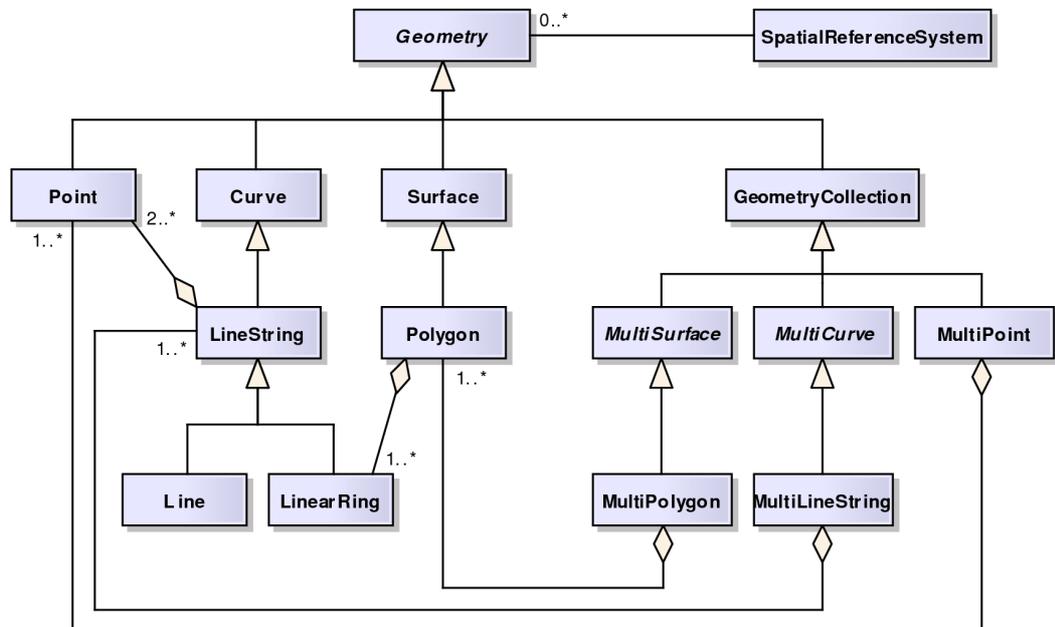


Abb. 3: Geometrieklassen des Simple-Feature-Modells (OGC 2005c)

Als Methoden für diese Geometrien stehen Basisfunktionen wie z. B. `GeometryType()`, die den Geometrietyp des Objekts zurück liefert, und Methoden zur Ermittlung topologischer Beziehungen zwischen Geobjekten auf Basis des „Dimensionally Extended Nine-Intersection Model“ (DE-9IM) bereit. Über diese Operationen können beispielsweise zwei Objekte auf geometrische Gleichheit (`Equals`) oder darauf geprüft werden, ob sie sich schneiden (`Intersects`). Des Weiteren stehen Methoden für räumliche Analysen wie z. B. „`Buffer`“ zur Erstellung einer Pufferzone um ein Geometrieobjekt oder „`SymDifference`“, die die symmetrische Differenz zweier Geometrien zurückgibt, zur Verfügung. Das Simple-Feature-Modell beschreibt zwei Repräsentationsformen für einfache Vektorgeometrien:

- Well-known Text (WKT) als lesbare Textrepräsentation
- Well-known Binary (WKB) als binäre Repräsentation

Zur Beschreibung von räumlichen Bezugssystemen wird vergleichbar eine WKT Repräsentation formuliert (siehe Kap. Räumliche Bezugssysteme).

Der SFA SQL-Teil spezifiziert die Implementierung des abstrakten Geometriemodells entweder über vordefinierte SQL Datentypen (numerische oder binäre Datentypen, als „normalized“ und „binary“ Geometrieschema bezeichnet) oder als objektrelationales Konzept über nutzerdefinierte, räumliche Datentypen (Geometriotypen). Zur Verwaltung der Geometrien bzw. der Geometrietabellen in der Datenbank wird eine Informationsschema vorgegeben, das sich je nach Implementierungskonzept leicht unterscheidet. Gemeinsam ist, dass in der Tabelle `Geometry_Columns` Metadaten wie der Name der Geometrietabelle und -spalte, die ID des SRS und die Dimension abgelegt werden, und ebenso, dass die verfügbaren räumlichen Bezugssysteme des Geodatenbanksystems in der Tabelle `Spatial_Ref_Sys` beschrieben werden müssen (OGC 2005c & 2005d).

Das OGC hat im Jahre 2006 eine neue Version 1.2.0 der SFA Spezifikation veröffentlicht, die einige Erweiterungen gegenüber dem zuvor beschriebenen Modell, entsprechend der vorangegangenen Version 1.1.0 bzw. der aktuellen ISO-Norm 19125:2004, beinhaltet. Den Geobjekten können nun zusätzlich auch Z-Koordinaten und M-Werte für Messungen hinzugefügt werden. Die räumlichen Prozeduren müssen aber weiterhin nur den zweidimensionalen Raum berücksichtigen (echte 3D Geometrieoperationen können optional unterstützt werden). Die Methoden sollen weitestgehend gemäß der ISO-Norm SQL/MM Spatial (siehe folgendes Kapitel) definiert werden. Des Weiteren enthält das Dokument ein Konzept zur Speicherung von räumlich platzierten Beschriftungen (Annotation Text) (OGC 2006a & 2006b).

SQL/MM Spatial

Die Norm ISO/IEC 13249 „SQL multimedia and application packages“ ist eine Erweiterung zur SQL Norm und enthält ein Teilpaket „Part 3: Spatial“

(ISO/IEC 13249-3:2006 (SQL/MM Spatial)), das ein Datenbankschema für Geodaten beinhaltet. In SQL/MM Spatial wird ein Schema von Geometrieklassen und zugehörigen Methoden vergleichbar dem Simple-Feature Standard definiert. Die Punkte der Geobjekte können X-, Y-, und Z-Koordinaten sowie Messwerte aufnehmen.

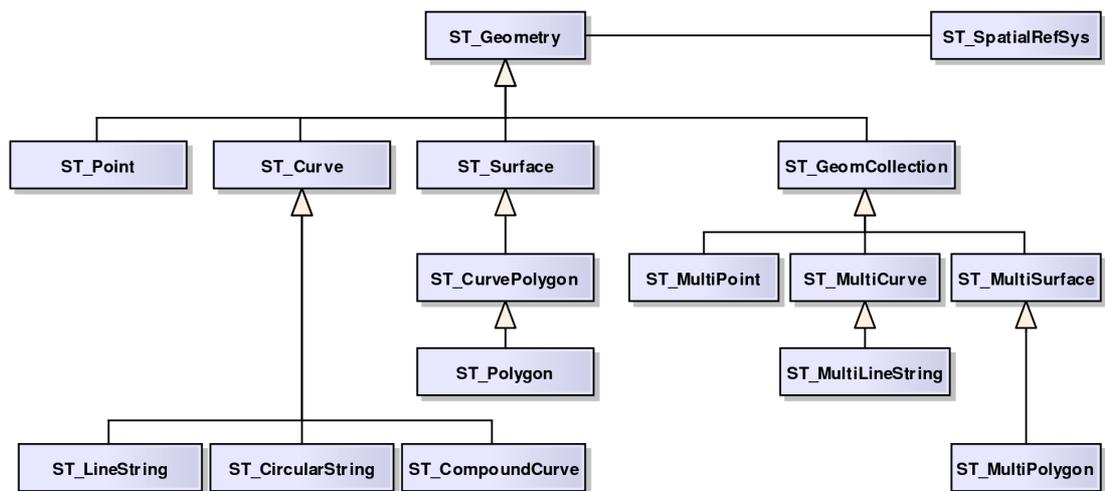


Abb. 4: SQL/MM Spatial Geometriemodell (Vom Autor verändert nach Brinkhoff 2008)

Im Unterschied zur SFA werden auch Kreisbögen für Linien und in zusammengesetzten Linienzügen oder Polygonen aus Geraden und Kreisbögen unterstützt. Als weitere Ergänzungen existieren u. a. Methoden zur Konstruktion aus oder der Konvertierung von Geometrien in die Geography Markup Language (GML), Prozeduren zur Validierung von Geometrien, Koordinatentransformation sowie Winkel- und Richtungstypen. Des Weiteren wird im Unterschied zu dem Simple-Feature Modell ein Datenbankschema für Netzwerkmodelle und Topologie beschrieben, das jedoch einfacher als im Spatial Schema (ISO 19107) modelliert ist. Weitere Vergleiche zu Geometrietypen und Funktionen zwischen Simple Feature Access und SQL/MM Spatial finden sich beispielsweise in SFA (OGC 2006b) oder Brinkhoff (2007b & 2008).

Die Norm SQL/MM Spatial spezifiziert als Implementierungsschema nur die Umsetzung mittels nutzerdefinierter, räumlicher Datentypen und entsprechende Konstruktoren für die Geometrie aus WKT, WKB oder GML. Ähnlich dem Simple Feature Schema werden Definitionen für Informationstabellen und Raumbezugssysteme angeboten (Brinkhoff 2007a & 2008).

Geography Markup Language

Der OGC Geography Markup Language (GML) Encoding Standard spezifiziert die XML-basierte Modellierung von geographischen Informationen und Anwendungsschemata. Mit GML wird eine offene, herstellerneutrale Beschreibungssprache für Geoobjekte zu deren Austausch und Speicherung in XML angeboten, die auf den konzeptuellen Modellen der ISO und OGC aufbaut. Durch die Abbildung des Spatial Schemas (ISO 190107) ist die Geography Markup Language sehr umfassend und komplex, weshalb der Standard die Erstellung von individuellen Profilen und Applikationsschemata vorsieht (OGC 2007). Ein solches Profil, das GML auf das Simple-Feature Modell beschränkt, wurde von der OGC als die „Geography Markup Language (GML) simple features profile“ Implementierungsspezifikation veröffentlicht (OGC 2006d).

Die mittlerweile sehr starke Verbreitung von XML in der IT als universell einsetzbares Datenformat zur Ablage und zum Austausch von Daten, vor allem auch im Kontext von Web-Applikationen und -Diensten, sowie die zunehmende Unterstützung von Datenbanksystemen zur Verwaltung von XML hat auch dazu geführt, dass GML bereits in der Geoinformationsverarbeitung etabliert ist (Yeung & Hall 2007). So setzt beispielsweise auch die OGC Web Feature Service Spezifikation (WFS) die Beschreibungssprache GML zum webbasierten Austausch von vektorialen Geometrien und Sachdaten von Geoobjekten zwischen Geoinformationssystemen über das Hypertext Transfer Protokolls (HTTP) obligatorisch ein (OGC 2005b). Seit der ersten Veröffentlichung von GML (als Empfehlung der OGC im Jahre 2000),

gab es allerdings mehrere inhaltliche Veränderungen und Weiterentwicklungen bis zum heutigen Standard GML 3.2.1 (seit 2007 auch als ISO Norm 19136 verabschiedet). Dieser Zustand spiegelt sich zum Teil noch in der Implementierung unterschiedlicher GML Versionen in verschiedenen Softwareprodukten oder im Grad der Umsetzung wieder, was den reibungslosen Umgang in der Praxis erschwert. In dem Datenbanksystem Oracle stehen z. B. für die Ausgabe von Geobjekten in GML oder deren Konstruktion aus GML entsprechende Funktionen für verschiedene GML Versionen bereit (allerdings mit der Einschränkung, dass nicht alle gültigen GML-Objekte unterstützt werden). Eine direkte räumliche Anfrage auf ein GML-Dokument ist nur über den Umweg einer redundanten Ablage als Geometriedatentyp in einer weiteren Tabelle möglich (Brinkhoff 2008).

Räumliche Bezugssysteme

Die Abbildung raumbezogener Objekte erfolgt in Geoinformationssystemen mittels räumlicher Referenzsysteme. Ein solches Raumbezugssystem besteht aus einem Bezugskörper, der die Erde darstellt, und einem Koordinatensystem, das eindeutig auf diesem verankert wird. Die Geometrie eines Geobjektes kann durch Koordinaten des Koordinatensystems beschrieben und somit exakt einer Position auf der Erde zugeordnet werden. Im Laufe der kartographischen Historie sind bis heute weltweit sehr viele, unterschiedliche Raumbezugssysteme, in Abhängigkeit des Verwendungszwecks und der globalen Gebietszuordnung, entstanden. Die ISO und das OGC haben in ihren Standards, wie beispielsweise Simple Feature Access oder Coordinate Transformation Services (OGC 2001), eine formale, textliche Beschreibung von räumlichen Referenzsystemen, die als „Well-known Text Representation of Spatial Reference Systems“ bezeichnet wird, definiert. Die Bezugssysteme werden dabei über verschiedene Attribute wie Bezugskörper, Datum, Projektion, Einheiten etc. spezifiziert. Mit der Definition der Raumbezugssysteme in WKT ist es möglich, diese einheitlich strukturiert in

Geodatenbanksystemen abzulegen und, wie in den zuvor dargestellten Geodatenmodellen beschrieben, jedem Geoobjekt genau ein solches räumliches Bezugssystem über eine Referenz (häufig als Spatial Reference ID (SRID) bezeichnet) zuzuweisen (OGC 2001 & 2006a).

An dieser Stelle ist es weiterhin wichtig zu erwähnen, dass das „Surveying and Positioning Committee“ der „International Association of Oil & Gas Producers“ (OGP) eine umfassende Dokumentation von Raumbezugssystemen („EPSG Geodetic Parameter Dataset“) mit Parametern zu Ellipsoiden, Datums, Koordinatensystemen etc. pflegt und offen zur Verfügung stellt. Diese Zusammenstellung wurde ursprünglich von der „European Petroleum Survey Group“ erstellt, die seit 2005 in das Surveying and Positioning Committee der OGP aufgegangen ist, deren Abkürzung EPSG allerdings bis heute als Bezeichnung der Bibliothek beibehalten wurde. So steht beispielsweise der EPSG-Code 4326 für geographische Koordinaten/Datum WGS 84 oder 31467 für deutsches Hauptdreiecksnetz (DHDN) Gauß-Krüger Zone 3 (OGP 2008). Diese Sammlung von Raumbezugssystemen hat sich mittlerweile in der Geoinformationsbranche so stark etabliert, dass nicht nur die meisten Software- und Datenhersteller diese Definitionen implementieren bzw. nutzen und zur Referenzierung eines Bezugssystems auf die entsprechende ID der EPSG Dokumentation (EPSG-Code) verwiesen wird, sondern dass auch einige Standards wie z. B. SFA oder WFS die Möglichkeit der Referenzangabe über den EPSG-Code spezifizieren.

2.4 Architekturen und Technologien

2.4.1 Softwarearchitekturen

Eine bedeutende Rolle bei der zunehmenden Vernetzung von verteilten Systemen zur gegenseitigen Informationsnutzung und -integration nimmt die

Softwarearchitektur ein. Die klassische Client-Server-Architektur bildet hierbei ein grundlegendes Modell für verteilte Softwaresysteme und stellt eine elementare Ausgangsbasis für komplexere Architekturen dar. In diesem einfachen Modell gibt es zwei Akteure: einen Server, der eine Ressource verwaltet, und einen Client, der die Ressource nutzen will. Zur Kommunikation sendet der Client eine Anfrage (Request) an den Server, der diese verarbeitet und die resultierende Antwort (Response) an den Client zurücksendet. Das Client-Server-Konzept ist in der Praxis gerade im Kontext von Datenbanksystemen weit verbreitet (siehe Kapitel 4.2.2) (Dunkel et al. 2008).

Der Einsatz von großen, komplexen Softwaresystemen erfordert allerdings eine Zerlegung in mehr als nur 2 Schichten, wie beim Client-Server-Modell (Zweischichtenarchitektur). Hierbei werden die Aufgaben eines Systems auf mehrere Schichten bzw. Komponenten verteilt. Die einzelnen Ebenen können gleichermaßen als Client und Server fungieren. Eine solcher Aufbau kann nach Dunkel et al. (2008) als N-Tier-Architektur bezeichnet werden. Solche mehrschichtige Architekturen werden insbesondere bei Web-Anwendungen eingesetzt.

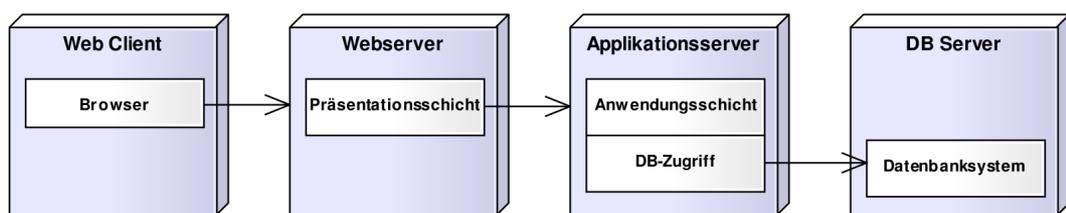


Abb. 5: Beispiel einer webbasierten N-Tier-Architektur (N=4) (Dunkel et al. 2008)

In den letzten Jahren ist nun der Begriff der „serviceorientierten Architekturen“ (service-oriented architectures (SOA)) zu einem regelrechten Hype in der IT-Branche geworden. Dieses Architekturkonzept entstand aus der Überlegung, dass eine erhöhte Nutzbarkeit von Anwendungen entstehen

würde, wenn die Komponenten über reine Daten- oder Service-Schnittstellen miteinander verbunden werden könnten, was vor allem die Integration von Applikationen in heterogenen IT-Landschaften zum Ziel hat. Beliebige Komponenten eines verteilten Systems werden dabei mit einer Programmierschnittstelle ausgestattet, die von beliebig anderen Komponenten im System aufgerufen werden können. Das SOA Konzept beschreibt eine allgemeine Architektur, deren Implementierung prinzipiell unabhängig von einer Technologie ist. Gegenwärtig erfolgt die technologische Realisierung häufig mittels Web Services (Dunkel et al. 2008).

2.4.2 Heterogenität

Moderne Architekturen wie SOA haben das Ziel die Heterogenität in bestehenden IT-Landschaften zu überwinden und eine bessere Interoperabilität der Systeme zu erreichen. Auch im Bereich der Geoinformationsverarbeitung herrschen heute in mittleren und großen Organisationen zum Teil heterogene GIS-Infrastrukturen vor, die sich durch unterschiedliche Systeme mit folglich unterschiedlichen Datenmodellen, Austauschformaten, Datenqualitäten, Standards, Prozessen und Verwaltungskonzepten charakterisieren. In der Entwicklung und im Ausbau von unternehmensweiten GIS-Architekturen zur gegenseitigen Geodatenutzung und -vernetzung sind die Anwendungen und Daten meist unter Berücksichtigung bestehender Strukturen zusammenzuführen und einzugliedern.

Im Kontext der allgemeinen Informationsintegration bezeichnen Leser und Naumann (2007) zwei Systeme als heterogen, wenn sie nicht die exakt gleichen Methoden, Modelle und Strukturen für den Zugriff auf ihre Daten anbieten. Die Facetten der Heterogenität sind dabei vielfältig und reichen von der technischen, syntaktischen, datenmodellbezogenen, strukturellen, schematischen bis zur semantischen Heterogenität. Im Zusammenhang mit (Geo-)Datenbanken differenzieren Yeung & Hall (2007) Heterogenität spezi-

fischer, wie folgende Abbildung verdeutlicht:

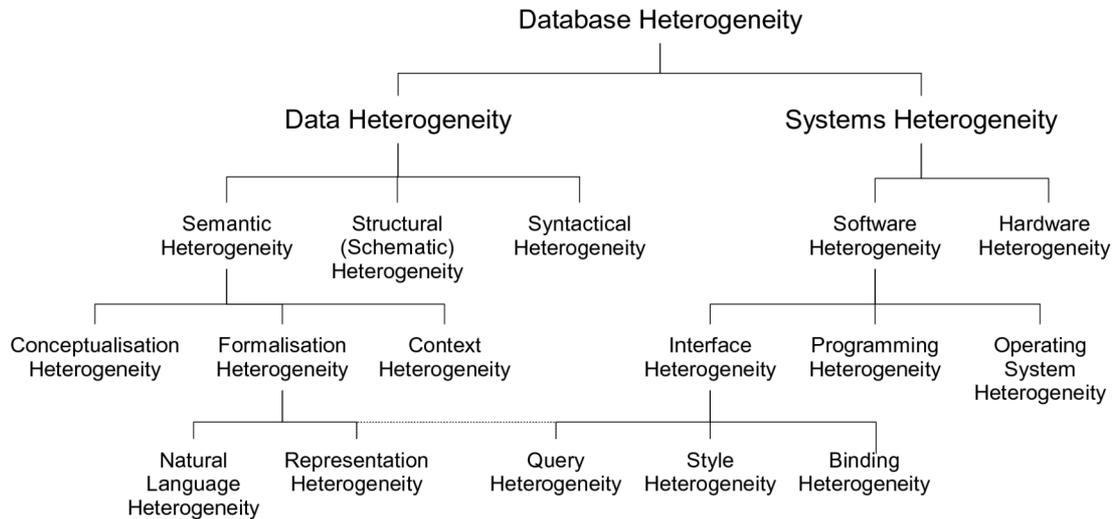


Abb. 6: Gliederung der Datenbank-Heterogenität (Yeung & Hall 2007)

Diese Arbeit beschäftigt sich mit der Heterogenität bezüglich Software, Schnittstellen (technisch), der Speicherung von Daten (strukturell, schematisch) und partiell auch mit der Geometriemodellierung (semantisch) im Kontext von Geodatenbanksystemen.

2.4.3 Nutzung verteilter Geodatenbanken

Zur interoperablen Nutzung und Integration von Geodaten in heterogenen Systemarchitekturen finden aktuell verschiedene Konzepte und Technologien ihre Anwendung, deren Verwendung sich immer stark an den Gegebenheiten und Anforderungen orientiert. Die grundlegenden, weit verbreiteten Verfahren werden in diesem Kapitel skizziert.

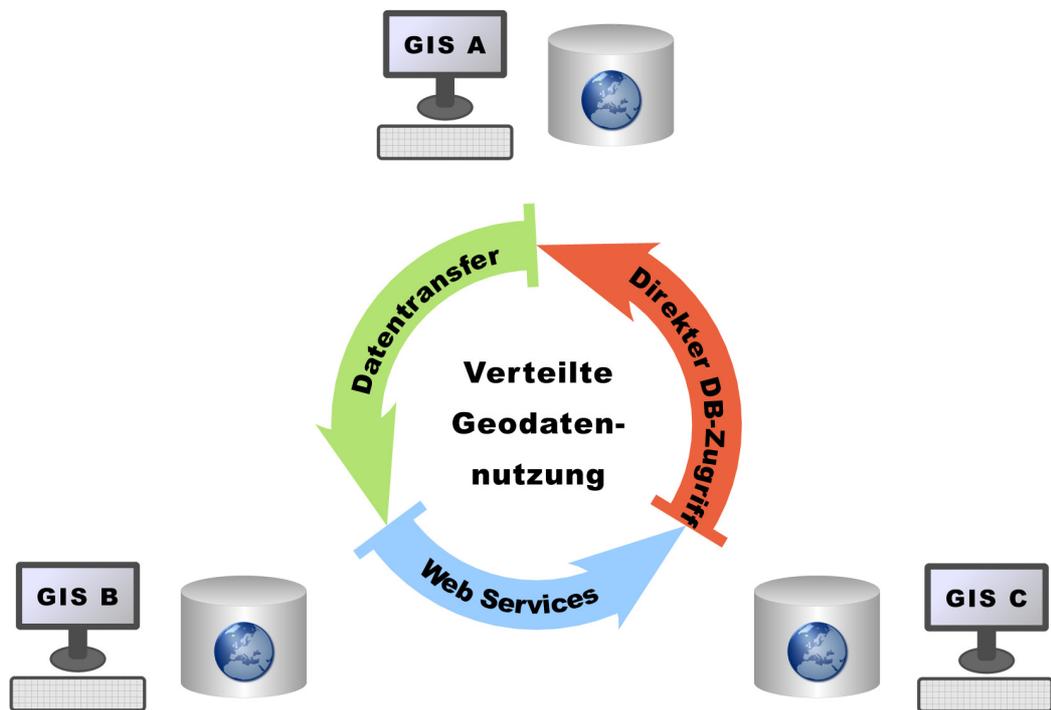


Abb. 7: Möglichkeiten zur Nutzung verteilter Geodatenbanken (Entwurf von Göttmann 2009)

Datentransfer

Ein erprobter, konventioneller Weg zur Nutzung verteilter, heterogener Geodatenquellen ist der Datentransfer. Die Geodaten werden dazu aus dem Quellsystem exportiert, transformiert und in das Zielsystem importiert. Der Schritt der Transformation bezieht sich häufig nur auf die Konvertierung in ein Austauschformat, welches durch das Zielsystem interpretiert werden kann. Neben den systemspezifischen Formaten kommen dabei auch de-facto-Standard-Formate wie beispielsweise das Shapefile Format des GIS Herstellers ESRI zum Einsatz. Der Transformationsschritt kann sich aber auch weiter untergliedern und beispielsweise die Veränderung der Georeferenzierung oder auch tiefer gehende Umwandlungen der Datenstruktur, Aggregationen oder komplette Modelltransformationen vornehmen. Eine weiterführende Vorstellung und Diskussion dieses Ansatzes, implizit seiner Leistungsfähigkeit, seiner Problematik, der Klassifikation von Transferverfahren und einer Gegenüberstellung zu Geodiensten, wurde von Donaubaue

(2004) unter der Bezeichnung der Geodatenintegration vorgenommen und soll an dieser Stelle nicht weiter vertieft werden.

Der Softwaremarkt offeriert gegenwärtig für den Datentransfer leistungsstarke Produkte oder Programmbibliotheken, die sich auf die Interpretation, Konvertierung und Transformation von Geodaten spezialisiert haben, wie beispielsweise die Feature Manipulation Engine (FME) des Softwareherstellers Safe Software Inc. (Safe 2009).

Im Datenbankumfeld werden analoge Verfahren auch als ETL Prozesse (Extract-Transform-Load) in Data-Warehouse-Systemen eingesetzt (Saake et al. 2008). In Anlehnung daran werden geodaten-spezifische Transferprozesse auch als Spatial ETL bezeichnet. Spatial ETL Prozesse dienen nicht ausschließlich zur einmaligen oder fallbezogenen Übertragung von Geodaten zwischen zwei oder mehreren Systemen, sondern finden ihren Einsatz auch häufig darin, wenn eine zentrale Geodatenhaltung, z. B. eine Auskunftsdatenbank eines Unternehmens (Geo-Data-Warehouse), aus verschiedenen Quellsystemen zyklisch und automatisiert geladen werden soll.

Bei einem Datentransfer findet folglich eine Materialisierung der Geodaten im Zielsystem statt. Das Zielsystem arbeitet also mit einer Kopie, deren Aktualitätsstand nur dem Zeitpunkt des Überführungsprozesses entspricht.

Webbasierte Geodienste

In mehrschichtigen Architekturkonzepten wie in den vorgestellten N-Tier-Architekturen oder modernen SOAs werden Zwischenschichten (Middleware) zwischen Datenbanken und Clients eingezogen und im Falle von SOA definierte Dienste bzw. Diensteschnittstellen angeboten. Im Umfeld der Geodatenverarbeitung stellt nach Donaubauer (2004, S. 22) ein Geodienst „über Schnittstellen Funktionalitäten für die Nutzung von Geodaten bereit. Die Nutzung von Geodaten umfasst den Zugriff auf sowie die Erfassung, Manipulation, Transformation, Analyse und die Präsentation von Geodaten“.

Die Realisierung von serviceorientierten Architekturen erfolgt häufig und in zunehmendem Maße über webbasierte Dienste (Web Services) (Dunkel et al. 2008). Das Open Geospatial Consortium hat für die Geoinformationsbranche verschiedene Spezifikationen solcher Web Services verfasst. Von diesen webbasierten Geodiensten gemäß den Schnittstellendefinitionen der OGC, sogenannte OGC Web Services (OWS), haben vor allem die Datendienste Web Map Service (WMS) und Web Feature Service (WFS) eine weite Verbreitung in den Softwareprodukten und in der Praxis gefunden. Ein WMS liefert Geodaten wie beispielsweise Karten oder Luftbilder als Rasterdaten über das HTTP Protokoll aus, wohingegen ein WFS vektorielle Daten bzw. die Geometrien im Austauschformat GML verfügbar macht (OGC 2005b, OGC 2006c, OGC 2009). Ein OGC Web Service oder sofern nicht durch die OGC spezifiziert, auch allgemein als Geo Web Service zu bezeichnen, ist somit eine Spezialisierung eines Geodienstes auf der Plattform der Web-Technologie (Donaubauer 2004). Die anfragende Anwendung benötigt bei einem Geodatenzugriff über einen Web Service keine Kenntnisse über die innere Struktur oder die spezifischen Schnittstellen und Ausgabeformate des Datenbanksystems, da sie nur mit der Middleware über die definierte und ggf. standardisierte Schnittstelle und nicht direkt mit dem Datenbanksystem kommuniziert.

Direkte Datenbankzugriffe

Ein weiterer Ansatz zur verteilten Geodatennutzung stellt der direkte Zugriff einer Anwendung auf die Datenhaltung eines oder mehrerer Fremdsysteme dar. Dabei kommt wieder das klassische Modell der Client-Server-Architektur zum Tragen und die Nutzung der Geodaten erfolgt über Dienste, die das Datenbankmanagementsystem bereitstellt und die zur Kommunikation die standardisierte Datenbanksprache SQL einsetzen. Dieser Ansatz ist Untersuchungsgegenstand dieser Arbeit und wird im folgenden Kapitel näher erläutert.

3 Lösungsansatz für den direkten Datenbankzugriff

In diesem Kapitel wird der theoretische Ansatz zur interoperablen Geodaten-nutzung über direkte Datenbankzugriffe hergeleitet und erläutert sowie die methodische Umsetzung innerhalb dieser Arbeit beschrieben.

3.1 Theoretischer Ansatz

Zur Herleitung des theoretischen Ansatzes wird an dieser Stelle zunächst nochmals die Zielsetzung der Master Thesis rekapituliert. Es sollen Erkenntnisse zur technischen Qualifizierung und dem strategischen Einsatz von direkten Datenbankzugriffen zu Geodatenbanksystemen für die interoperable Geodaten-nutzung in heterogenen GIS-Landschaften gewonnen werden. Diese Programmatik bedingt die Untersuchung und Diskussion des Ansatzes über direkte Datenbankzugriffe. Die direkte Kommunikation zwischen Client und Datenbank ist kein neues, sondern bewährtes Konzept, dessen Umsetzung im Bereich der Geoinformationsverarbeitung unter Berücksichtigung der aktuellen Standardisierungen und Implementierungen hier aufgegriffen wird. Die Betrachtung schließt dabei die existierenden Modelle und Technologien ein, analysiert die technischen Voraussetzungen, eruiert mögliche Einsatzbereiche und stellt Überlegungen an, wie sich die Integration dieses Ansatzes in eine IT- respektive GIS-Landschaft vollziehen kann.

Die Architekturen von Datenbanksystemen sind ein in der Praxis weit verbreiteter Anwendungsfall für Client-Server-Modelle. Das Datenbanksystem stellt dabei Dienste bereit, die die Abfrage und Manipulation der Inhalte der Datenbank durch einen Client ermöglichen. Die Kommunikation findet üblicherweise über die standardisierte Schnittstelle bzw. Abfragesprache SQL

statt (Dunkel et al. 2008).

Handelt es sich bei den Daten nun um Geodaten, kann die im vorausgegangenen Kapitel vorgestellte Definition eines Geodienstes herangezogen werden. Das OGC hat bereits 2002 eine Spezifikation für eine OpenGIS Service Architecture herausgegeben (OGC 2002), die später in die ISO Norm 19119 überging. Dieser Standard definiert Architekturen für Diensteschnittstellen, die für geographische Informationen genutzt werden können. Als technologische Plattformen⁴ zur Zusammenarbeit von Systemen werden neben der bereits bekannten Web-Technologie (Web Services) auch weitere wie beispielsweise Java 2 Enterprise Edition (J2EE), Common Object Request Broker Architecture (CORBA) und auch SQL spezifiziert. Für GI-Managementdienste (Feature access service), die einem Client Zugang zu und die Verwaltungsmöglichkeit über eine Geodatenhaltung anbieten, werden u. a. die ISO Normen 19107 und 19125 als relevant angesehen, die um die Norm ISO/IEC 13249-3 ergänzt werden können.

Als Ausgangsbasis lässt sich aus den bisherigen Grundlagen Folgendes festhalten:

- Normen spezifizieren die Modellierung von Geodaten, die Analyse ihrer topologischen Beziehungen und den Zugriff über die standardisierte Abfragesprache SQL
- Clients und Geodatenbanksysteme kommunizieren in einer Client-Server-Architektur direkt, indem das DBS Dienste zur Abfrage und Manipulation von Geodaten über die SQL Schnittstelle bereitstellt
- Geoinformationssysteme speichern ihre Geodaten zunehmend in Geodatenbanksystemen und implementieren entsprechende Konzepte und Zugriffsschnittstellen

Aus diesen Erkenntnissen lässt sich die Überlegung ableiten, dass unter

⁴ Distributed Computing Platform (DCP)

diesen Voraussetzungen auch der direkte Datenbankzugriff von unabhängigen Clients auf verteilte, heterogene Geodatenbanken möglich sein und als eine Form der interoperablen Geodatennutzung angesehen werden kann.

Als theoretischer Ausgangspunkt wird daraus die Hypothese aufgestellt, dass der Ansatz der direkten Datenbankzugriffe unter bestimmten Voraussetzungen und für spezielle Anforderungen, vor allem innerhalb einer Organisation bzw. eines Unternehmensnetzwerkes, als ein adäquates Verfahren für die interoperable Geodatennutzung in heterogenen (GI-)Systemarchitekturen eingesetzt werden kann.

Im Rahmen dieser Arbeit wird ein direkter Datenbankzugriff zur interoperablen Geodatennutzung wie folgt definiert:

Ein direkter Datenbankzugriff ist ein Zugriff eines Geoinformations- oder sonstigen Clients auf die Geodaten in einem Geodatenbanksystem via SQL, ohne eine zusätzliche Zwischenschicht, die die Daten strukturell modifiziert, transformiert oder materialisiert.

Der Ansatz kann sich aufgrund verschiedener technischer Möglichkeiten und unterschiedlicher Arten der beteiligten Systeme dahingehend unterscheiden, dass die interoperable Geodatennutzung sowohl Geoinformations- oder primär Nicht-GIS-Applikationen als auch Verbindungen von Datenbanken untereinander mit einbezieht.

3.2 Methoden

Die Methodik beschreibt wie dieser Ansatz innerhalb der hier vorliegenden Arbeit anhand einer Empirie eruiert wird. Sie stützt sich zunächst auf eine eingehende Studie und Analyse der aktuell verfügbaren, sowohl standardisierten als auch proprietären Modelle, Technologien und Konzepte. Die

praktische Untersuchung des Ansatzes bzw. der Hypothese erfolgt über eine exemplarische Testumgebung auf Basis verschiedener Geodatenbank- und Geoinformationssysteme, anhand derer spezifizierte Testfälle durchlaufen werden. Die technisch-orientierte Empirie dient zur Evaluation des theoretischen Ansatzes und zur Ableitung von verallgemeinerten Aussagen der wissenschaftlichen Untersuchung.

Zur Qualifizierung der existierenden Modelle und Technologien für den direkten Datenbankzugriff zur interoperablen Geodatennutzung wurden verschiedene Indikatoren herangezogen und innerhalb der Testumgebung praktisch überprüft. Folgende Indikatoren wurden herangezogen:

- die Implementierung der Geodatenverwaltung in den DBS
- die Definition, Konstruktion und der Zugriff auf Geoobjekten via SQL
- die möglichen Datenbankzugriffsschnittstellen und Verbindungsmethoden der GIS-Clients
- der Umgang mit einfachen Geometrien (Simple Features)
- der Umgang mit komplexen Geometrien
- räumlich platzierte Beschriftungen
- die Kombination, Konfiguration und das Rechtekmanagement von Objektklassen aus heterogenen Datenbanksystemen
- Programmierschnittstellen und Technologien zur virtuellen Datenintegration

4 Direkter Datenbankzugriff in heterogenen GIS-Architekturen

Im Hauptteil dieser Master Thesis wird die ausgewählte Testumgebung und die Testverfahren für die praktische Untersuchung der direkten Zugriffsmöglichkeiten auf Geodatenbanksysteme von heterogenen GIS-Softwareprodukten und -komponenten dargestellt.

4.1 Testumgebung für direkte Datenbankzugriffe

Zur Qualifizierung der existierenden Modelle und Technologien für den direkten Zugriff auf Geodatenbanksysteme in einer heterogenen Systemlandschaft wurde eine Testumgebung aufgebaut, anhand derer verschiedene Indikatoren in praktischen Testfällen durchlaufen wurden.

4.1.1 Testarchitektur

Die Vielfalt der vorhandenen Softwareprodukte im Bereich der Geodatenbanksysteme und der (Desktop) Geoinformationssysteme einerseits und der Rahmen sowie die Ziele dieser wissenschaftlichen Untersuchung andererseits schließen eine ganzheitliche Analyse aus. Die praktische Durchführung erfolgt demnach an einer exemplarischen Testumgebung mit einer stark beschränkten Anzahl an Softwareprodukten. Die Softwareauswahl muss sich dabei an den aktuellen Entwicklungen, Technologien und dem praktischen Einsatz orientieren, um ein möglichst realistisches Verfahren zu gewährleisten und allgemeine Aussagen aus den gewonnenen Ergebnissen zu zulassen. Aus der umfangreichen Palette der am Markt existierenden Systeme wurden zwei Datenbanksysteme, die über eine Erweiterung für Geodaten verfügen, und drei Desktop Geoinformationssysteme für die Testarchitektur ausgewählt. Die Zusammenstellung enthält gezielt sowohl

Vertreter aus dem kommerziellen als auch aus dem Open-Source Bereich. Für detaillierte Produktbeschreibungen wird an dieser Stelle auf die Produktdokumentationen der Hersteller verwiesen. Die folgende Abbildung zeigt den generellen Aufbau der Testumgebung:

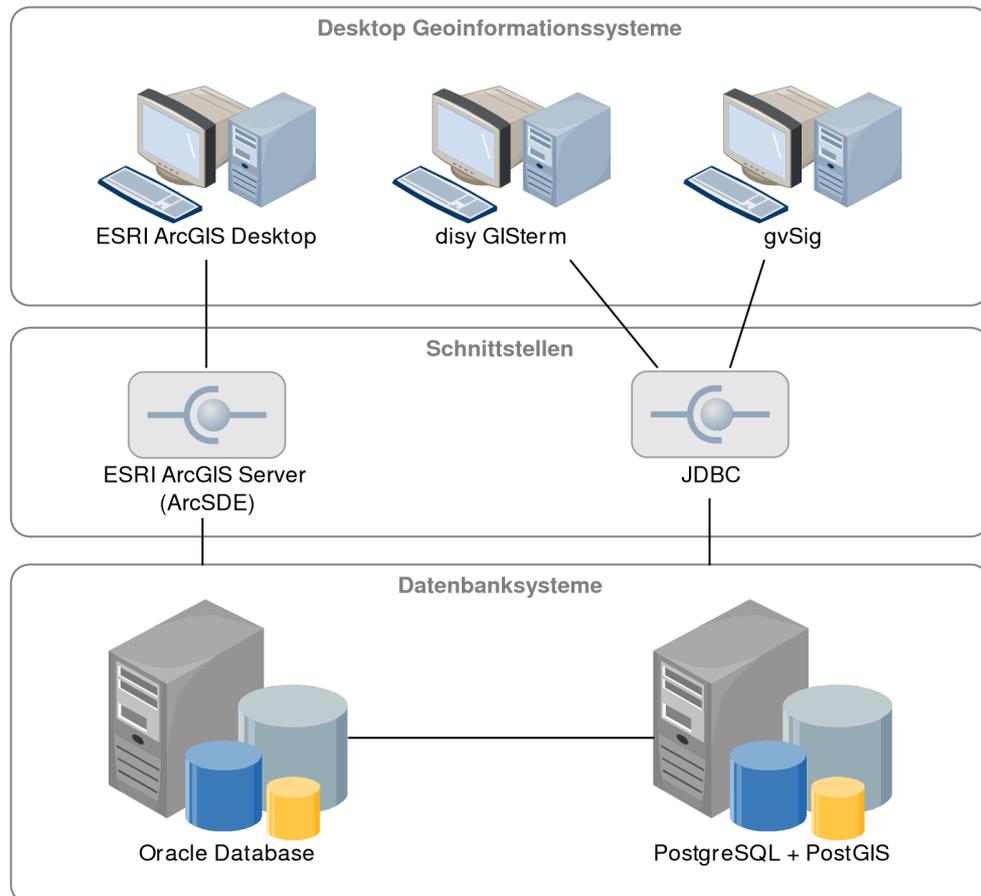


Abb. 8: Software und Architektur der Testumgebung (Entwurf von Göttmann 2009)

Aus den sechs im Kapitel 2.3.1 vorgestellten Datenbanksystemen, wurde die Software Oracle Database als Vertreter der kommerziellen Produkte und PostgreSQL mit der Erweiterung PostGIS als Repräsentant der Open-Source-Community ausgewählt. Beide Systeme stellen in ihrem jeweiligen Lizenzbereich derzeit den wohl größten Funktionalitätsumfang bereit (vgl. Brinkhoff 2007b).

Oracle Database

Das objektrelationale Datenbanksystem Oracle Database wird von dem Softwarehersteller Oracle Corporation entwickelt und stellt umfassende Funktionalitäten für das Datenmanagement, vor allem auch für große Unternehmen und IT-Architekturen, bereit. Die Oracle Database wird in verschiedenen Ausführungen mit unterschiedlichem Funktionsumfang lizenziert. Die Option Oracle Spatial bietet für die höchste Produktvariante Oracle Database Enterprise Edition eine Sammlung von Funktionen und Prozeduren zur Speicherung, dem Zugriff und der Analyse von Geodaten an. In der einfacheren Ausführung Standard Edition ist der sogenannte Oracle Locator enthalten, der nur eine Kernmenge an Funktionen aus dem Gesamtumfang von Oracle Spatial unterstützt. In der Testarchitektur wurde die Oracle Database Enterprise Edition mit Spatial Option in der aktuellen Version 11g (Release 1) und partiell auch in der Vorgängerversion 10g (Release 2) eingesetzt.

PostgreSQL und PostGIS

Das objektrelationale Datenbanksystem PostgreSQL, das ursprünglich aus einer Datenbankentwicklung der University of California in Berkeley entstand, wird im Bereich der Geoinformationsverarbeitung in der Regel mit der Erweiterung PostGIS, die ein Simple-Feature-konformes Klassenmodell und umfangreiche Funktionen bereitstellt, verwendet. Diese Kombination weist unter den Vertretern der Open-Source-Datenbanksysteme aktuell den größten Funktionalitätsumfang zur Verwaltung und Speicherung von Geodaten auf und wurde somit auch für die Testumgebung gewählt. Daneben stellt PostgreSQL auch native Geometrietypen bereit, die allerdings nicht konform zu den einschlägigen Normen und Standards sind und nur eingeschränkte Funktionalitäten bieten (Brinkhoff 2007b), weshalb diese für das Testverfahren nicht berücksichtigt wurden. PostgreSQL ist unter der BSD Lizenz und PostGIS unter der im Open-Source-Bereich weitläufig gebräuch-

lichen GNU General Public License verfügbar, die jeweils die freie Verwendung, auch im kommerziellen Geschäftsumfeld, erlauben. In der Testarchitektur wurde PostgreSQL in der Version 8.3 (und anteilig auch in der Vorgängerversion 8.2) mit der Erweiterung PostGIS 1.3 eingesetzt. Die Erweiterung PostGIS, die von dem kanadischen Unternehmen Refractions Research entwickelt und gepflegt wird, wurde von der OGC als konform zur Spezifikation SFA 1.1 Part 2 zertifiziert und sie implementiert auch teilweise die ISO Norm SQL/MM Spatial (PostgreSQL 2008, PostGIS 2008a).

Der Markt an geoinformationsverarbeitender Software bietet heute eine nahezu nicht überschaubare Vielfalt an Geoinformationssystemen für den Desktop-, Server- oder Mobil-Bereich an. Ein umfassender Evaluierungsprozess konnte der Testarchitektur somit nicht zugrunde gelegt werden. Die Softwareentscheidung orientierte sich an den Kriterien Einsatzzweck, Herkunft, Datenbankunterstützung und Technologie. Die Testumgebung beschränkt sich auf Desktop GIS-Software, die sowohl aus dem proprietären als auch aus dem quelloffenen Umfeld stammt. Als Grundvoraussetzung mussten die beiden eingesetzten Datenbanksysteme unterstützt werden. Die funktionale Ausstattung der GIS-Technologie war primär nicht entscheidend, sollte aber die Bandbreite von der einfachen bis hin zur High-End-Software widerspiegeln. Die nachfolgende Beschreibung gibt Erläuterungen zu den eingesetzten Softwareprodukten und den Auswahlgründen.

ESRI ArcGIS Desktop und ArcGIS Server

Aus der Produktfamilie ArcGIS des renommierten, amerikanischen Softwareherstellers für Geoinformationssysteme, der Environmental Systems Research Institute Inc. (ESRI), wurde die Software ArcGIS Desktop und Server eingesetzt. ArcGIS Desktop ist ein Desktop Geoinformationssystem zur Erzeugung, Analyse und Visualisierung von raumbezogenen Daten. Die Softwarereihe setzt sich aus verschiedenen Produktvarianten mit differen-

ziertem Funktionalitätsumfang sowie -erweiterungen für spezielle Anforderungen zusammen, aus der die Ausführung ArcEditor 9.3 und die Erweiterung ArcGIS Data Interoperability verwendet wurde. Die Data Interoperability Erweiterung ist eine Interoperabilitätsschnittstelle für ArcGIS, die den direkten, lesenden Zugriff oder den Import-Export von nicht ESRI-eigenen Geodatenformaten aus ArcGIS Desktop ermöglicht. ArcGIS Server fungiert als eine Serversoftware, die Geodaten oder geoinformationsverarbeitende Funktionalitäten über Dienste an Clients in verteilten und mehrschichtigen Architekturen ausliefert. Das Serverprodukt integriert neben der eigentlichen Serverkomponente ein spezielles Datenmanagementmodul, die Arc Spatial Database Engine (ArcSDE). Diese Technologie ermöglicht die GeodatenSpeicherung und -verwaltung der ESRI ArcGIS Produktfamilie (ESRI Geodatabase) in objektrelationalen Datenbankmanagementsystemen. Aus der Serverreihe kam die Variante ArcGIS Server Basic 9.3 zum Einsatz, von der nur die ArcSDE Komponente verwendet wurde (Esri 2009).

Disy GISterm

Das Produkt GISterm des Softwareunternehmens disy Informationssysteme GmbH ist ein Desktop Geoinformationssystem zum Erfassen, Verwalten, Analysieren und Visualisieren von räumlichen Daten. Das System setzt kein eigenes Datenformat zur Speicherung der Geodaten ein, sondern unterstützt über entsprechende Schnittstellen die Verwaltung und Ablage sowie den Zugriff auf räumliche Daten in Datenbanksystemen, u. a. Oracle Database und PostgreSQL. Die Software wurde in der Version GISterm Desktop 3.9 eingesetzt (disy 2007).

GvSIG

Die Open-Source-Software gvSig wurde von dem Amt für Infrastruktur und Transport (CIT) der spanischen, autonomen Region Valencia ins Leben gerufen und wird mittlerweile von einer offenen, internationalen Gemeinschaft von Entwicklern und Anwendern weiter vorangetrieben. Das quellof-

fene, javabasierte Werkzeug verfügt über die grundlegenden Eigenschaften eines Desktop GIS und implementiert, neben einigen OGC Spezifikationen, auch Datenbankschnittstellen für den Zugriff auf die verwendeten Geodatenbanksysteme. Zum Einsatz kam die Version gvSig 1.1, die unter der GNU General Public License veröffentlicht wird (gvSig 2007).

4.1.2 Geodaten

Das Testverfahren untersucht den allgemeinen, technischen Zugriff bzw. die Zugriffsschnittstellen auf Geodatenbanksysteme und das Lesen der Geometrieobjekte durch die Desktop Software unter Berücksichtigung der Standardisierungen von ISO und OGC sowie nicht standardisierten Ausprägungen. Die Eignung oder Notwendigkeit zur Verwendung eines konkreten Testdatensatzes oder -gebietes ist nicht gegeben. Vielmehr könnte der Einsatz von existierenden Geodaten, die in die Datenbanksysteme importiert werden müssten, eine Fehlerquelle aufgrund von bestehenden Geometrie- oder Formatfehlern darstellen. Als Indikatoren werden in den einzelnen Testfällen die verschiedenen Geometrieklassen der Standards sowie nicht standardisierte Sonderfälle durchlaufen. Für diese werden folglich standard- oder herstellerkonforme Musterdatensätze generiert, um die Aussagekraft und Sicherheit der Ergebnisse zu gewährleisten.

4.2 Testverfahren

Im Folgenden wird das Testverfahren beschrieben, dessen Schema sich in Indikatoren, darauf angewandte Methoden und den auftretenden Ereignissen gliedert. Im ersten Schritt wird die generelle Implementierung der Geodatenverwaltung in den Datenbanksystemen betrachtet. Die sich anschließende Testreihe durchläuft grundlegende Geometrieklassen der Standards, bevor relevante Anwendungs- und Sonderfälle untersucht werden.

4.2.1 Implementierung der Geodatenverwaltung in den Datenbanksystemen

Oracle Database

Die räumliche Erweiterung Oracle Spatial des Oracle DBMS unterstützt das objektrelationale Modell zur Repräsentation von Objektgeometrien. Die Vektordaten werden dabei in dem Oracle-eigenen, räumlichen Datentyp `SDO_GEOMETRY` abgelegt. Die Datenspeicherung korrespondiert somit zur Implementierung „SQL with Geometry Types“ der OGC SFA (Oracle 2006 & 2007) und ist in der Version Oracle Spatial 10g von der OGC als „compliant“⁵ zu SFA Part: 2 SQL Types and Functions 1.1 zertifiziert (OGC 2009). Die Klassen gemäß der ISO Norm SQL/MM Spatial existieren undokumentiert bereits seit der Version 10g, dokumentiert werden diese aber erst ab Release 11.1 (Brinkhoff 2008).

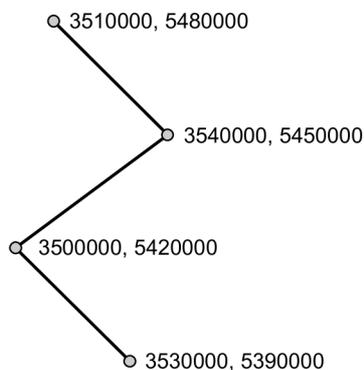
Der Geometriedatentyp `SDO_GEOMETRY` ermöglicht die Speicherung des geometrischen Attributs eines Geoobjektes in einer Spalte der Datenbankta-
belle. Er ist wie folgt definiert:

```
CREATE TYPE sdo_geometry AS OBJECT (  
  SDO_GTYPE NUMBER,  
  SDO_SRID NUMBER,  
  SDO_POINT SDO_POINT_TYPE,  
  SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY,  
  SDO_ORDINATES SDO_ORDINATE_ARRAY);
```

Das Attribut `SDO_GTYPE` gibt den Geometrietyp (z. B. Punkt oder Linie) über eine vierstellige Ziffer an. Der Identifikator des räumlichen Bezugssystems wird in `SDO_SRID` angegeben. Der Schlüssel referenziert auf eine Definition eines Raumbezugssystems in der Tabelle `SDO_COORD_REF_SYS`, die alle verfügbaren Bezugssysteme enthält. Sofern es sich um ein Punktobjekt handelt, können im Attribut `SDO_POINT` die X-, Y- und Z-Koordinaten gespeichert werden. Oracle empfiehlt für eine effiziente Beschreibung von

⁵ Der OGC Zertifizierungsgrad „compliant“ bedeutet, dass das Produkt das OGC Compliance Testing Program durchlaufen hat und alle verpflichtenden Elemente erfüllt (OGC 2009).

einfachen Punkten dieses Attribut zu verwenden. Daneben können Punkte aber auch wie andere Geometrien über die Attribute `SDO_ELEM_INFO` und `SDO_ORDINATES` beschrieben werden. Das Zahlenfeld `SDO_ORDINATES` speichert alle Koordinaten einer Geometrie. Das Attribut `SDO_ELEM_INFO` gibt an, wie diese Koordinaten zu interpretieren sind. Nähere Erläuterungen zur Struktur des Datentyps finden sich in Oracle (2007) und Kothuri et al. (2004). Das folgende Beispiel zeigt eine einfache Linie in `SDO_GEOMETRY` im Raumbezugssystem DHDN Gauß-Krüger Zone 3 (EPSG Code: 31467):



```
SELECT geom FROM linie;
```

```
-----
SDO_GEOMETRY(2002, 31467, NULL,
SDO_ELEM_INFO_ARRAY(1,2,1),
SDO_ORDINATE_ARRAY(3530000, 5390000,
3500000, 5420000, 3540000, 5450000,
3510000, 5480000))
```

Abb. 9: Beispiel einer Linie in `SDO_GEOMETRY`

Jedes Attribut vom Typ `SDO_GEOMETRY` benötigt in Oracle Spatial beschreibende Metadaten in einem Informationsschema, die in der Tabelle `SDO_GEOM_METADATA_TABLE` des Datenbankschemas `MDSYS` abgelegt werden. Für Datenbankbenutzer stehen verschiedene Sichten (sogenannte Views) auf diese Tabelle bereit. Der View `USER_SDO_GEOM_METADATA` zeigt die benutzerspezifischen Metadaten. Ein Metadateneintrag besteht aus vier Attributen: dem Tabellennamen, dem Namen der Geometriespalte, einer Beschreibung der Dimensionen und einer ID des Raumbezugssystems (Oracle 2007).

Ein `SDO_GEOMETRY` Objekt kann über die Oracle-eigenen Konstruktoren

direkt in der SDO-Struktur oder aus WKT, WKB oder GML erstellt werden. Der Zugriff auf die Geometrien kann, neben der allgemeinen Ausgabe des nativen Datentyps, auch über Zugriffsmethoden in WKT, WKB und GML erfolgen (siehe Tabelle 2 im Anhang).

Neben dem Oracle-eigenen Objekttyp SDO_GEOMETRY existiert auch die Klasse ST_GEOMETRY gemäß der Norm ISO SQL/MM Spatial. Diese Klasse kann als sogenannte „Wrapper-Klasse“ bezeichnet werden, da sie nur über ein Attribut vom Typ SDO_GEOMETRY definiert wird:

```
ST_GEOMETRY (geom SDO_GEOMETRY)
```

Die Klasse dient in Oracle einerseits zur partiellen Unterstützung von SQL/MM Spatial und andererseits auch zur Erfüllung der Konformität der SFA. Es werden Synonyme bereitgestellt, die SFA-Geometrietypen auf die Klassen von ST_GEOMETRY abbilden (Brinkhoff 2008). Die Konstruktion von Geometrien ist über Methoden der SQL/MM Spatial Norm möglich.

PostgreSQL und PostGIS

In einem PostgreSQL DBMS mit der Erweiterung PostGIS werden die Geometrien in einer PostGIS-eigenen, erweiterten Form von WKT und WKB der OGC SFA 1.1.0 (als PostGIS EWKT bzw. EWKB bezeichnet) repräsentiert, indem zusammen mit den Koordinaten auch der Identifikator des Raumbezugssystems abgelegt wird. Die Speicherung der Geometrien erfolgt im PostGIS Geometry Datentyp (nachfolgend als PG_GEOMETRY bezeichnet). Ein Select Statement liefert für die gleiche Linie wie im obigen Beispiel bei einer ASCII Ausgabe das kanonische Standardformat des PostGIS Geometrietyps als EWKB in Hexadezimalform (HEXEWKB):

```
SELECT geom FROM linie;
```

```
-----
```

```
0102000020EB7A0000040000006666660688EE4A4133333303AC8F5441666666  
06F0B34A4133333303F8AC54416666660610024B413333330344CA5441666666  
0678C74A413333330390E75441
```

Zur Erstellung eines Geoobjektes bietet PostGIS zahlreiche Konstruktoren gemäß der OGC SFA, der ISO/IEC Norm SQL/MM Spatial sowie eigene Funktionen an. In PostgreSQL/PostGIS werden alle Geometrien im Typ PG_GEOMETRY abgelegt, unabhängig davon, von welchem Konstruktor sie erzeugt wurden. Für den Zugriff auf die Geometrien stehen, neben der Standardausgabe, auch Zugriffsmethoden bereit (siehe Tab. 2).

Das Informationsschema besteht aus der Tabelle GEOMETRY_COLUMNS, die die Tabelleninformationen, die Geometriespalte, die Koordinatendimension, die SRID und die Geometrieklasse erfasst, und der Tabelle SPATIAL_REF_SYS, die die Informationen über die räumlichen Bezugssysteme enthält (PostGIS 2008b). PostGIS wurde in der Version 1.1 bei der OGC als „compliant“ zur Simple Feature Spezifikation SQL 1.1 Types and Functions registriert (OGC 2009).

ESRI ArcGIS Server (ArcSDE)

Die Softwarekomponente ArcSDE der ESRI Software ArcGIS Server ist kein Datenbanksystem, sondern eine Datenbankschnittstelle, die von ESRI Clients für den direkten Zugriff sowie die Verwaltung und Speicherung des ESRI Geodatenmodells in Datenbanksystemen benötigt wird. Zusätzlich ermöglicht das Modul innerhalb einer ESRI Softwarelandschaft auch Verwaltungsfunktionen für den gleichzeitigen Mehrbenutzerbetrieb, das Management von langen Transaktionen und Versionen, die Replikation in verteilten Datenbankumgebungen und die Historisierung von Datenbeständen.

Die physische Speicherung der Geometriedaten erfolgt über ArcSDE in Abhängigkeit der eingesetzten Datenbanksoftware respektive deren unterstützten Datentypen. In Oracle und PostgreSQL wird bei der Installation der ArcSDE Komponente ein ESRI-eigener, nutzerdefinierter Datentyp, ESRI ST_GEOMETRY, im Datenbankschema SDE erstellt, der zur OGC SFA Part 2 und zur ISO/IEC 13249-3 SQL/MM Spatial korrespondiert. Das Klassenmodell orientiert sich dabei weitestgehend am Geometrieschema der

ISO Norm SQL/MM Spatial. Daneben können auch die nativen Speicherformate der räumlichen Datenbankeerweiterungen, wie Oracle SDO_GEOMETRY oder PostGIS PG_GEOMETRY, genutzt werden. Bei der Verwendung des Oracle Datenbanksystems mit ArcSDE können des Weiteren binäre Speicherformen, die die Geometrien im Datentyp Binary Large Object ablegen, wie das ESRI-eigene ArcSDE Compressed Binary oder OGC WKB, verwendet werden (Esri 2008a). ArcSDE 9.3 ist für Oracle und PostgreSQL OGC „compliant“ zu SFA SQL 1.1 Types and Functions (OGC 2009).

Aus den internen Speichermodellen wird ersichtlich, dass bei der Verwendung der ESRI Software in Kombination mit einer Oracle Database folglich der Datentyp ST_GEOMETRY zweifach, d. h. sowohl als Oracle als auch als ESRI Datentyp vorhanden ist. Die Struktur dieser Typen ist aber völlig verschieden. Um hier Namenskonflikten vorzubeugen, müssen in Oracle durch andere Systeme implementierte ST_GEOMETRY Datentypen über ein Schemapräfix angesprochen werden. Ansonsten wird der Oracle-eigene Datentyp bei der Spaltendefinition verwendet (Oracle 2007).

Definition, Konstruktion und Zugriff

Die Normen ISO/IEC 13249-3 und ISO 19125-2 bzw. die OGC SFA - Part 2: SQL Option definieren eine SQL Spracherweiterung für räumliche Daten. Dieser Abschnitt beschäftigt sich mit dem allgemeinen Zugriff via SQL auf die Geodaten in den Datenbanksystemen Oracle und PostgreSQL/PostGIS unter Berücksichtigung dieser Standards und den räumlichen Datenbankeerweiterungen. Es werden dazu die Definitionen und die verfügbaren Konstrukturen und Zugriffsmethoden der räumlichen Datentypen betrachtet und differenziert. Die praktische Untersuchung wurde für Oracle mittels der SQL Clients sqlplus und SQL Developer und für PostgreSQL über psql und pgAdmin durchgeführt. Die Tabellen 1 und 2 im Anhang geben eine Übersicht⁶ über die Definition, Konstruktion und die Ausgabe der räumli-

⁶ Die Übersicht ist ein Auszug und stellt die wesentlichen Methoden der Typen dar. Zur vollständigen Dokumentation sind die Softwareprodukte und die Benutzerhandbücher heranzuziehen.

chen Datentypen. Aus der Synthese der bisher gelegten Grundlagen und der hier erfolgten Empirie können an dieser Stelle folgende Aussagen für den weiteren Testverlauf festgehalten werden:

- *Die relevanten Standards geben Implementierungsmodelle für die Geometrien mittels vordefinierter (numerische oder binäre, nur SFA) oder nutzerdefinierter Datentypen vor. Die Struktur der nutzerdefinierten Geometriedatentypen respektive die interne Speicherung von Instanzen der Geometrieklassen wird durch die Standards aber nicht festgelegt. Die berücksichtigten Systeme verwenden jeweils eigene, räumliche Datentypen bzw. Speicherformate.*
- *Zur Konstruktion und dem Zugriff auf die Geometrien finden sich in den räumlichen Erweiterungen der Systeme unterschiedliche Konstruktoren und Zugriffsmethoden, die aus verschiedenen Versionen der SQL/MM Spatial und Simple Feature Access Standards stammen, die nach diesen individuell verändert oder gänzlich als eigene Funktionen implementiert wurden (siehe Tabelle 2 im Anhang).*
- *Die Geometrieerstellung und der Zugriff sowie die Abfrage von topologischen Beziehungen ist generell via SQL möglich. Dabei sind aber die systemspezifischen, räumlichen Implementierungen sowie die Umsetzungen der relevanten Standards zu berücksichtigen. Dies erschwert oder beschränkt zum Teil den praktischen Umgang.*
- *In den Softwaredokumentationen finden sich meist nur allgemeine Bezüge zu den Standards, d. h. Angaben zur genauen Version oder dem Grad der Konformität fehlen, sind u. U. aber sehr entscheidend.*

4.2.2 Lesende Datenbankzugriffe über heterogene GIS-Clients

Die folgenden Testfälle (Indikatoren) untersuchen die praktische Umsetz-

barkeit des direkten Zugriffs auf die Datenbanksysteme von heterogenen GI-Softwareprodukten und liefern somit einen wesentlichen Teil der wissenschaftlichen Empirie zum theoretischen Ansatz aus Kapitel 3.

Zugriffsschnittstellen

Der erste Indikator dient zur Ermittlung der Art und Weise der Datenbankverbindung, die zwischen den GIS-Clients und den Datenbanksystemen aufgebaut wird. Die Desktop Applikationen GISterm und gvSig stellen für den Zugriff eine JDBC Verbindung zu den Datenbanksystemen her. Die Software ArcGIS Desktop verwendet die Komponente ArcSDE, um auf die Datenbank zuzugreifen. Diese kann von dem Client entweder über einen ArcSDE Service oder über einen ArcSDE DirectConnect genutzt werden. Im ersten Fall fungiert ArcSDE als Applikationsserver, der dem Client einen Dienst bereitstellt. Beim DirectConnect, der in den Testfällen eingesetzt wurde, wird eine direkte Verbindung zwischen Client und Datenbanksystem aufgebaut. Der DirectConnect auf eine Oracle Datenbank benötigt und verläuft über die Oracle Client Software.

Mit Ausnahme des ArcSDE Services, hier kann man von einer Dreischichtenarchitektur sprechen, handelt es sich um zweischichtige Softwarearchitekturen, bei denen die Anwendungslogik vollständig im Client abläuft (auch als Fat Client bezeichnet, vgl. Dunkel et al. 2008). Der Client muss entsprechend leistungsstark sein, da er die Benutzeroberfläche (Präsentationsschicht), die fachlichen Funktionalitäten (Anwendungsschicht) und den Datenbankzugriff realisiert.

Für den generellen Aufbau einer Verbindung zwischen Client und Datenbanksystem werden die Zugangsinformationen zur Datenbank wie die Adresse des Datenbankservers und des Ports, der Name der Datenbank sowie ein Datenbankbenutzer und Passwort benötigt. Nach der Anmeldung am Datenbanksystem über ein entsprechendes Menü der GIS-Software, stehen die Tabellen oder Datensichten zur Auswahl, für die der Benutzer auto-

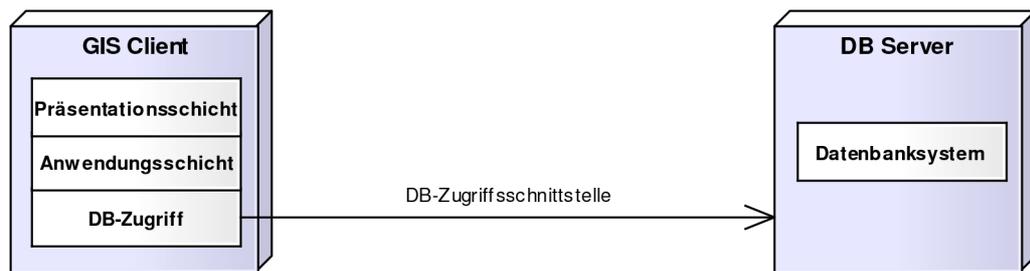


Abb. 10: Zweischichtige GIS-DB-Architektur (Vom Autor verändert nach Dunkel et al. 2008)

risiert ist. Der Anwender erhält eine direkte Sicht auf das Datenbankschema und muss folglich die inhaltliche Strukturierung der Daten kennen. Die Datenbanktabellen mit einem geometrischen Attribut können als Ebene im Kartenfenster des GIS-Clients visualisiert werden. Dazu beziehen die Clients die Daten via SQL aus dem Geodatenbanksystem. Es wird dabei in der SQL-Abfrage meist ein räumlicher Filter verwendet, damit nur die Daten gelesen und zum Client transportiert werden müssen, die auch im aktuellen Kartenausschnitt zu sehen sind.

Die Anwendungen nutzen die nativen Speicherformen der Datenbanksysteme (oder ggf. auch die GIS-eigenen Datentypen, z. B. ESRI ST_GEOMETRY). Die standardisierten Repräsentationsformen werden i. d. R. nicht direkt verwendet. Die Interpretation der individuellen Geometrieausgaben erfolgt über entsprechende Programmbibliotheken bzw. -module der Desktop Geoinformationssysteme.

In der ArcGIS Produktfamilie müssen die Geodatentabellen, die nicht über die Software selbst in der Datenbank erstellt wurden (sondern durch andere Applikationen), zuvor explizit in der Datenbankschnittstelle ArcSDE registriert werden, damit sie in einem ArcGIS Client verwendet werden können.

- *Die Verbindungen zwischen Client und Datenbanksystem in der Zweischichtenarchitektur werden über standardbasierte (z. B. JDBC) oder herstellerspezifische Datenbankschnittstellen aufgebaut.*

- *Zum Verbindungsaufbau werden Zugangsinformationen zur Authentifizierung am Datenbanksystem benötigt. Der Anwender erhält eine direkte Sicht auf das für ihn autorisierte Datenbankschema.*
- *Die Desktop Geoinformationssysteme nutzen i. d. R. die systemspezifischen, räumlichen Speicher- bzw. Ausgabeformen. Die Geometrien werden via SQL ausgelesen und über entsprechende Programmbibliotheken interpretiert.*

Simple Features

Die ISO Norm 19125 bzw. die OGC SFA Spezifikation definieren einfache Geometrien (Simple Features) für Punkte, Linien und Flächen (Polygone). Diese geometrischen Grundformen sollten als minimale Vereinbarung in einer heterogenen Systemlandschaft untereinander ausgetauscht und genutzt werden können. Die entsprechenden Testfälle untersuchen einerseits ob und wie die Simple Feature Geometrien in den Geodatenbanksystemen modelliert werden können und andererseits den lesenden Zugriff durch die GIS-Clients. Die Testdaten für die einfachen Geometrien wurden gemäß den Konstruktoren der Geodatenbanksysteme über SQL, sowohl aus WKT als auch in den systemspezifischen Strukturen, erstellt. Als Speicherformat wurden jeweils die nativen, räumlichen Datentypen der Datenbanksysteme (SDO_GEOMETRY bzw. PG_GEOMETRY) genutzt.

Die Geometrieklassen Punkt, Linie, einfacher/nicht einfacher/einfacher, geschlossener und nicht einfacher, geschlossener Linienzug, Polygon und Polygon mit Loch sind durch die Desktop Systeme der Testarchitektur in den spezifischen Geometriedatentypen lesbar. Die aktuelle OGC SFA Spezifikation in der Version 1.2.0 enthält eine zusätzliche, neue Geometrieklasse Polyhedral Surface⁷, die im Rahmen dieser Studie unberücksichtigt blieb.

⁷ Kontinuierliche Ansammlung von Polygonen mit gemeinsamen Grenzen (OGC 2006a)

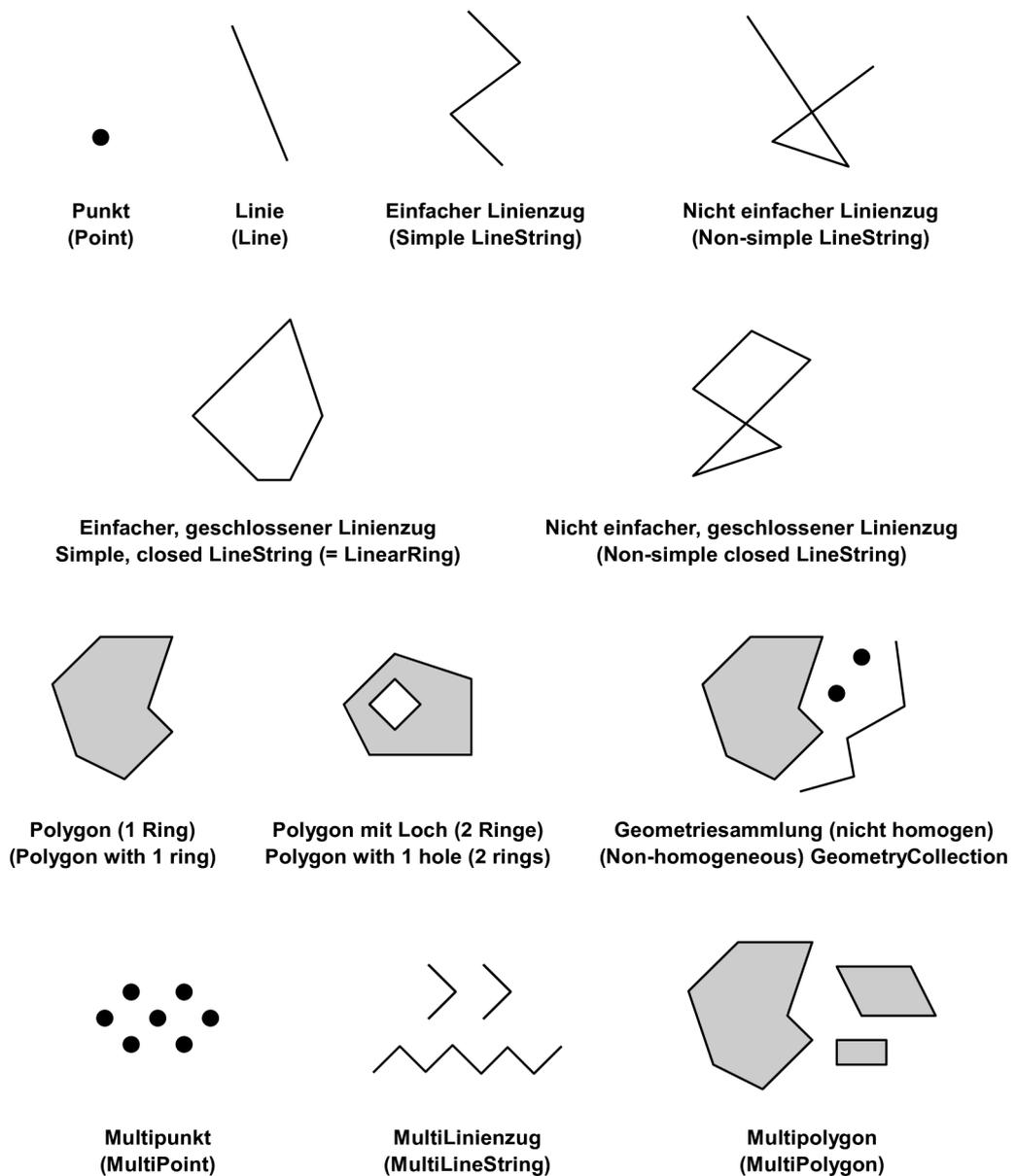


Abb. 11: Beispiele für Simple Feature Geometrien (Entwurf von Göttmann 2009)

Das Modell der Simple Feature Geometrieklassen weist eine Unterklasse GeometryCollection aus, die sich aus heterogenen oder im Speziellen nur aus homogenen Einzelgeometrien zusammensetzen kann (vgl. Kap. 2.3.2). Die homogenen Geometriesammlungen (MultiPoint, MultiCurve, MultiSurface), werden durch die GIS-Clients interpretiert. Die heterogenen Zusammenstellung von Geometrien verursachen allerdings Zugriffsprobleme. Geo-

objekte, die aus verschiedenen Geometrieklassen bestehen, können durch die Systeme ArcGIS Desktop und GISterm nicht gelesen werden, unabhängig von der eingesetzten Datenbanksoftware. Dieses Verhalten tritt gleichermaßen bei unterschiedlichen Geometrieklassen innerhalb eines geometrischen Attributs (Tabellenspalte) auf. Diese Relationen können in mehrere Views, die nur eine Geometrieklasse enthalten, zerlegt werden. Diese Sichten sind dann problemlos durch die Desktop Programme lesbar. Das System gvSig liest GeometryCollections auch ohne den Umweg über Views, allerdings sind die grafischen Formatierungsmöglichkeiten im Vergleich zu den Einzelklassen eingeschränkt.

Ein weiterer Testfall prüft den Umgang mit mehreren geometrischen Attributen in einer Datenbanktabelle. Der unmittelbare Zugriff auf die Geodaten ist nur mit gvSig möglich, da bei der Definition der zu lesenden Datenbankrelation die Geometriespalte explizit ausgewählt werden kann. Bei den beiden anderen Systemen ist aufgrund unterschiedlicher Einschränkungen wiederum eine Umgehungslösung über Sichten erforderlich.

- *Die Simple Feature Geometrieklassen können in den systemspezifischen Geometriedatentypen der Geodatenbanksysteme modelliert und von den Desktop Geoinformationssystemen weitestgehend interpretiert und visualisiert werden.*
- *Das Verhalten der GIS-Clients bei der Interpretation von heterogenen GeometryCollections ist softwareabhängig. Eine Zerlegung in die einzelnen Geometrieklassen mittels Datenbanksichten ermöglicht aber meist die Anzeige.*

Kreisbögen und Kurven

Das Simple Feature Geometriemodell definiert nur lineare Verbindungen von Stützpunkten. Die Norm ISO/IEC 13249-3 sieht im Gegensatz dazu auch die Verwendung von Kreisbögen in linie- oder flächenhaften Geoobjek-

ten vor. Die nativen, räumlichen Implementierungen der Datenbanksysteme Oracle und PostgreSQL/PostGIS stellen Konstruktoren für Kreisbögen bereit. Die Testfälle bilden einzelne Kreisbögen, Linienzüge und Polygone aus Kreisbögen und Geraden sowie Kreise ab. Zur Konstruktion dienen entsprechend die Geometrieklassen CircularString, CompoundCurve und CurvePolygon (bei Oracle als Arc LineString, Compound LineString, Compound Polygon und Circle bezeichnet) nach der ISO Norm SQL/MM Spatial.

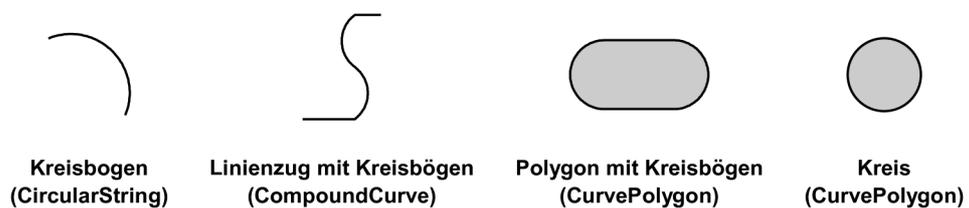


Abb. 12: Beispiele für Geometrien mit Kreisbögen (Entwurf von Göttmann 2009)

Die Interpretation der Kreisbögen durch die Desktop Software fällt sehr unterschiedlich aus. Die Anzeige von Kreisbögen bzw. der zusammengesetzten Objekte aus Kreisbögen und Geraden aus der PostgreSQL/PostGIS Datenbank ist keinem der Systeme möglich. Bei der Speicherung in Oracle werden die Kreisbögen-Objekte durch die Clients softwareabhängig entweder korrekt, fehlerhaft oder auch gar nicht interpretiert.

Ein weitere, nicht lineare Verbindung von Punkten ist über weiche, elegante Kurven (Bézier-Kurven, B-Splines etc.) möglich. Diese speziellen Abbildungen der Vektorgraphik sind weder in den Standards zu Simple Features oder in SQL/MM Spatial noch in den nativen Geometriemodellen bzw. Speicherformaten der Datenbanksysteme zu finden. Nur im konzeptuellen Datenmodell der Norm ISO 19107 sind solche Verbindungen vorgesehen.

- *Die Modellierung und Speicherung von Kreisbögen in den Geodatenbanksystemen ist generell möglich. Die Interpretation durch die GIS-*

Clients ist allerdings sehr unterschiedlich bzw. softwareabhängig.

- *Die Verbindung von Stützpunkten über elegante Kurven (Splines) ist weder in den relevanten Modellen der Normen und Standards (Ausnahme: ISO 19107) noch in den Datenbanksystemen zu finden.*

Beschriftungen

Die Normen ISO 19125 bzw. OGC SFA (Version 1.1.0) und ISO/IEC 13249-3 SQL/MM Spatial enthalten keine Definition für räumlich platzierte Beschriftung von Geoobjekten (auch Kartenbeschriftung, Annotations, Labels genannt). Die aktuelle Version 1.2.0 der OGC SFA Spezifikation gibt hier ein Lösungskonzept „Annotation Text“ für einfach Beschriftungsobjekte vor, das bewusst nicht darauf ausgerichtet ist, hohen kartographischen Qualitätsansprüchen zu genügen, aber eine Annäherung an diese bietet (vgl. OGC 2006a, S. 42-52). Diese Definition ist im Oracle DBMS ab der Version 11g implementiert, aber nur rudimentär dokumentiert. Das georeferenzierte Textobjekt wird im Oracle ST_ANNOTATION_TEXT Datentyp abgelegt. Die Formatierung (Schriftart, -größe etc.) wird dabei in einem Textattribut im XML Schema gemäß OGC SFA spezifiziert. Die Erstellung einer Beschriftung über SQL war praktisch durchführbar. Der spezielle Beschriftungsdatentyp wird allerdings noch von keinem der verwendeten Desktop GIS unterstützt. In PostgreSQL/PostGIS existiert noch keine Umsetzung für georeferenzierte Beschriftungen.

Die ESRI ArcGIS Software verfügt über ein eigenes Konzept zur Verwaltung und Speicherung von Beschriftungen, das größtenteils offen implementiert und somit für andere Systeme lesbar ist. In einer sogenannten Annotation Feature Class werden Textobjekt mit einer Geometrie (= Umrandungspolygon des Textes) und entsprechenden Styleattributen (Schriftart, -größe, -auszeichnung, Drehwinkel etc.) abgelegt. Inwieweit diese Informationen durch andere Anwendungen verarbeitet bzw. genutzt werden können, ist

von den jeweiligen Layoutmöglichkeiten abhängig. Das Auslesen des Drehwinkels aus einem Attribut der Beschriftungstabelle und dessen Verwendung zur Darstellung ist bei den getesteten Softwaresystemen möglich.

Hinsichtlich der räumlich platzierten Beschriftungen besteht noch ein Interoperabilitätsdefizit. Es existieren keine oder nur proprietäre Umsetzungen (OGC 2006a). Die Implementierung des Modells der OGC SFA 1.2.0 ist, zumindest in den getesteten Systemen, gegenwärtig noch nicht praxistauglich fortgeschritten.

Sachdaten

Das Lesen von nicht geometrischen Attributen (Sachdaten) eines Geoobjektes ist, sofern die Geometriedaten im Desktop System grundsätzlich angezeigt werden können, generell möglich. Es können Einschränkungen hinsichtlich der Interpretation oder inkorrekte Anzeigen, wie beispielsweise die Dezimaldarstellung einer ganzen Zahl aus Oracle vom Typ number(x) in gvSig, auftreten. Im Rahmen dieses Testfalls wurden allerdings nur einfache Text- und Zahlendatentypen betrachtet. Das Verhalten bei sonstigen SQL- oder herstellerspezifischen Datentypen wurde nicht untersucht.

Kombination von Objektklassen

Die Kombination von Geodaten aus unterschiedlichen Datenhaltungen stellt eine Grundanforderung für die Nutzung verteilter Datenquellen dar. Die Zusammenführung mehrerer, verschiedener Objektklassen in einer Anwendung ist in allen eingesetzten Desktop Geoinformationssystemen uneingeschränkt durchführbar, sofern die Geodatentabellen prinzipiell visualisiert werden können. Dabei spielte es keine Rolle, ob die Daten nur aus einem oder aus beiden, verwendeten Geodatenbankprodukten stammten.

Benutzersichten und Objektberechtigungen

Ein Datenbankzugriff in heterogenen Landschaften und Organisations-

strukturen durch Anwender respektive Systeme, die nicht die datenbesitzenden Stellen bzw. die Quellsysteme sind, geschieht oftmals über benutzerspezifische Datenansichten. Der Datenbesitzer kann damit differenziert kontrollieren, welche Daten dem anderen Datennutzer (z. B. einer anderen Fachabteilung) zur Verfügung stehen. Vom Datenanbieter müssen über Datenbankwerkzeuge entsprechende Sichten und Objektberechtigungen eingerichtet werden.

Die Testfälle konnten nachweisen, dass Datenbanksichten auf Tabellen mit geometrischen Attributen von den Desktop GIS gleichermaßen wie die materialisierten Basistabellen genutzt werden können, sofern die Sicht dem Datenbankbenutzer gehört, der auch die zugrundeliegende Tabelle besitzt.

Die Vergabe einer Leseberechtigung durch einen Besitzer (u1) einer Tabelle oder eines Views an einen anderen Benutzer (u2) reicht für ArcGIS Desktop und GISterm, im Gegensatz zu gvSig, nicht zur Anzeige durch den Anwender u2 aus (SQL Beispiel: `GRANT SELECT ON TABLE geotabelle TO u2`). Der Zugriff auf die Daten ist nur dann möglich, wenn die Sicht auf die Daten von u1 auch dem Benutzer u2 gehört. In ArcGIS muss der GRANT-Befehl zusätzlich mit dem Parameter „WITH GRANT OPTION“ erteilt werden, ansonsten lässt sich die Tabelle nicht in ArcSDE registrieren. Diese Aktion hat allerdings die Nebenwirkung, dass der Berechtigte (u2) nun auch die Berechtigung auf dieses Datenbankobjekt an andere Benutzer weitergeben darf (ggf. stellt dies eine Sicherheitslücke dar). Neben dem Erzeugen eines Views durch den Benutzer u2 muss dieser auch die Informationstabellen in seinem Datenbankschema füllen, d. h. die Metadaten für die Sicht mit geometrischem Attribut ergänzen. Im Oracle DBMS steht jedem Benutzer automatisch eine eigene Sicht auf das Informationsschema der Geotabellen zur Verfügung. In der PostgreSQL/PostGIS Datenbank hingegen müssen die Tabellen `Geometry_Columns` und `Spatial_Ref_Sys` im jeweiligen Benutzerschema zunächst generiert werden.

- *Die Anzeige von Geodaten über Datenbanksichten wird von den Desktop GIS generell unterstützt. Die im Datenbankumfeld gebräuchliche Vergabe einer SELECT Berechtigung durch den Besitzer einer Tabelle oder eines Views an einen anderen Benutzer ist zur Visualisierung von Tabellen mit geometrischem Attribut softwareabhängig nicht immer ausreichend. In diesen Fällen muss die Sicht im Schema des anderen Benutzers erstellt und ggf. weiterführende Objektberechtigungen, die zu Sicherheitseinschränkungen führen können, erteilt werden.*
- *Der Besitzer der Sicht muss die Metainformationen des Geodaten-Views in seinem Datenbankschema ergänzen.*

ST_GEOMETRY

Die Norm ISO/IEC 13249-3 SQL/MM Spatial definiert ein Geometrieschema mit der Oberklasse ST_Geometry (vgl. Abb. 4). Zur Erzeugung von Geoobjekten stehen ST-Methoden für die Unterklassen bereit (z. B. ST_Point). Dieser Testfall soll klären, wie Desktop Systeme mit den Geometrien umgehen, die über die ST-Konstrukturen mittels SQL in den Datenbanksystemen erstellt wurden.

Das DBMS Oracle bildet die ST_Geometry Klasse als „Wrapper-Klasse“ auf der Klasse SDO_Geometry ab (vgl. Kap. 4.2.1). Die Tabellen mit geometrischem Attribut vom Datentyp Oracle ST_Geometry respektive die Geometrien sind durch keines der Desktop Produkte interpretierbar.

In PostgreSQL/PostGIS werden alle Geometrien im Datentyp PG_Geometry abgelegt, unabhängig von welchem Konstruktor (impliziert auch die Methoden von SQL/MM Spatial) sie erzeugt wurden, und sind somit für die Desktop GI-Systeme lesbar.

Wie bereits unter 4.2.1 erwähnt, implementiert die ESRI Komponente ArcSDE bei der Installation eine ESRI-eigene ST_Geometry Klasse im Benutzerschema SDE der Datenbanksysteme. Geometrien, die explizit über die

ESRI ST-Konstrukturen generiert werden, sind bezüglich der Datentypstruktur folglich absolut verschieden zu den Geometrien, die über die datenbankeigenen ST-Konstrukturen abgelegt wurden. Die Interpretation des ESRI ST_Geometry Datentyps wird folglich nur durch die ArcGIS Produkte selbst unterstützt. Sofern die ESRI ST-Geometrien in PostgreSQL vorliegen, kann ein Workaround⁸ über einen View, der implizit eine Datentypumwandlung von ESRI ST_Geometry zu PG_Geometry vornimmt, dennoch das Lesen über die beiden anderen Systeme ermöglichen, ohne dass die Daten zusätzlich materialisiert werden müssen. In der Oracle Datenbank funktioniert dies nicht.

Das folgende Beispiel zeigt die jeweilige Erstellung und Standardausgabe eines Punktes über die ST-Funktionen. Die Norm ISO/IEC 13249-3 definiert zur Konstruktion eines Punktes u. a. die Methoden: ST_Point(x,y,srid) oder ST_PointFromText(wkt,srid).

```
Oracle: ST_Point(3500000.00, 5400000.00, 31467)
```

```
ST_GEOMETRY(SDO_GEOMETRY(2001, 31467, SDO_POINT_TYPE(3500000,  
5400000, NULL), NULL, NULL))
```

```
PostGIS: ST_PointFromText('POINT(3500000.00 5400000.00)', 31467)
```

```
0101000020EB7A000000000000F0B34A410000000070995441
```

```
ESRI: SDE.ST_Point(3500000.00, 5400000.00, 2)
```

```
ST_GEOMETRY(1, 1, 3500000, 5400000, 3500000, 5400000, NULL, NULL, NULL,  
NULL, 0, 0, 2, '0C0000000100000080A9ABF4A20380EAB0BBFB08')
```

Die Konstruktion von Geometrien gemäß den Methoden der Norm ISO/IEC 13249-3 (SQL/MM Spatial) stellt keineswegs sicher, dass die Objekte durch die Desktop GI-Systeme gelesen werden können. Die interne Definition der ST_Geometry Klasse ist jeweils verschieden und deren Interpretation ist softwareabhängig.

⁸ Umgehungslösung oder Hilfskonstruktion

Repräsentation durch WKT, WKB und GML

Die Konstruktion und Ausgabe von Geometrien in den Repräsentations- und Austauschformaten WKT, WKB und GML, die in den Standards Simple Feature Access und Geography Markup Language spezifiziert sind, ist grundsätzlich über entsprechende Funktionen der Datenbanksysteme möglich (siehe Tabelle 2 im Anhang). In PostgreSQL/PostGIS stehen in der eingesetzten Version für die Konstruktion von Geometrien aus GML noch keine Methoden bereit. Die Desktop Geoinformationssysteme visualisieren die geometrischen Attribute, sofern diese in den nativen, räumlichen Datentypen abgelegt sind. Geometrien, die nur in den standardisierten Repräsentationsformen WKT, WKB oder GML verfügbar sind, können aus der Datenbank nicht direkt gelesen werden. Nur die ESRI ArcSDE Software in Kombination mit dem Oracle Datenbanksystem unterstützt auch das Format OGC Well-Known Binary. Dieser Testfall soll untersuchen, ob Geometriedaten, die in den standardisierten Repräsentationsformen vorliegen bzw. bezogen werden, in die datenbankeigenen Formate umgewandelt und somit in den Desktop GIS angezeigt werden können, ohne dass eine weitere Materialisierung der Daten erfolgen muss.

In der Oracle Datenbank wurden dazu Geometrien als WKT in einem Attribut abgelegt (Datentyp varchar) und darauf eine Sicht eingerichtet, die die WKT Daten als SDO_Geometry Objekte anzeigt, z. B.:

```
CREATE VIEW wkt2sdo_v AS SELECT id, SDO_UTIL.FROM_WKTGEOMETRY(wkt)
AS "SDOGEOM" FROM wkttabelle;
```

Eine SQL Select auf diesen View zeigt korrekte SDO_Geometry Daten an. Sofern entsprechende Metadaten für diese Sicht erstellt werden, liegt die Annahme nahe, dass diese Sicht wie eine ursprünglich in SDO_Geometry gespeicherte Geometrietabelle behandelt werden kann. Die Verwendung in den GIS-Clients widerlegt diese Annahme.

Die Systeme stellen für die Anzeige der Geodaten im Kartenfenster eine SQL-Anfrage mit einer räumlichen Filtermethode an die Datenbank. Diese Abfrage benötigt einen räumlichen Index (Spatial Index) auf die Geometriespalte. Die Indexierung eines Views ist allerdings nicht durchführbar, weshalb die Anwendungen den fehlenden Spatial Index monieren.

Ein weiterer Ansatz versucht eine eigendefinierte Funktion, die aus WKT SDO_Geometry Objekte erzeugt, anstelle der vordefinierten FROM_WKT-GEOMETRY(wkt) Methode in der Sicht zu verwenden, so dass die räumliche Indexierung über einen sogenannten „Function-Based Spatial Index“ vorgenommen werden kann. Mit dieser Technologie wird der Spatial Index auf die Funktion erstellt, d. h. die Daten müssen nicht explizit materialisiert werden (Kothuri et al. 2004). Um in Oracle eine Funktion zu indexieren, muss diese allerdings als „deterministic“ spezifiziert werden. Eine determinierte Funktion ist funktional bewusst so konzipiert, dass sie bei mehrmaligem Aufruf mit den gleichen Parametern nur einmal ausgeführt wird. Dieses Verhalten resultiert in einem unerwünschten Effekt für diesen speziellen Anwendungsfall, wie das folgende Beispiel verdeutlicht:

Die Spalte sdogeom ist die Spalte des Views, die durch die eigendefinierte, indexierte Funktion (hier: wkt2sdo(wkt)) repräsentiert wird. Dieser Funktion wird die Spalte (hier: wkt) mit den WKT Daten übergeben.

```
CREATE VIEW wkt2sdo_v AS SELECT id, wkt2sdo(wkt) AS "sdogeom" FROM
wkttabelle;
```

Im SQL Statement zur Datenabfrage wird für den räumlichen Filter eine Bedingung eingesetzt, so dass die Funktion zweifach aufgerufen wird:

```
SELECT id, sdogeom FROM wkt2sdo_v WHERE SDO_RELATE(sdogeom,
MDSYS.SDO_GEOMETRY(2003, 31467, NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,
1003,3),MDSYS.SDO_ORDINATE_ARRAY(3300000,5300000,3600000,5500000)),
'mask=anyinteract querytype=WINDOW') = 'TRUE';
```

Das Ergebnis des SQL Statements liefert somit nur NULL-Werte für sdo-

geom und nicht, wie gewünscht, SDO_Geometry Geometrien. Die Anzeige durch die Desktop Systeme ist folglich nicht möglich. Weitere Lösungsansätze wurden im Rahmen dieser Arbeit nicht eruiert.

Im DBMS PostgreSQL führte die Erstellung einer Sicht, die aus einem WKT Textattribut mit der Methode ST_GeomFromText(wkt,srid) PostGIS Geometrieobjekte erzeugt, zum gewünschten Ergebnis. Die Views können wie Tabellen gelesen werden, obwohl die Daten materialisiert nur als WKT in einem Textdatentyp vorliegen. Dazu müssen die Metadaten des Views im Informationsschema, aber nicht explizit ein Spatial Index, vorhanden sein.

- *Die Konstruktion und Ausgabe von Geometrien via SQL in den Repräsentations- und Austauschformaten WKT, WKB und GML ist weitestgehend möglich.*
- *Die Desktop GIS unterstützen generell die systemspezifischen, räumlichen Datentypen. Geometrien, die in den standardisierten Repräsentationsformen verfügbar sind, können nicht unmittelbar zur Anzeige von Geoobjekten in den Clients verwendet werden. In PostgreSQL/PostGIS ist dies über eine Sicht mit impliziter Typumwandlung ohne zusätzliche Materialisierung der Daten möglich. Für das DBMS Oracle konnte ein entsprechender Ansatz nicht gefunden werden, da die eingesetzten Systeme zwingend eine räumliche Indexierung zur Datenabfrage benötigen.*

Interoperabilitätsadapter

In der Geoinformationsverarbeitung existiert eine Vielzahl an verschiedenen Geodatenformaten unterschiedlichster Ausprägung und Ursprungs. Dies wird annähernd deutlich, wenn man beispielsweise die umfangreiche Konvertierungs- und Transformationssoftware für Geodaten FME betrachtet (siehe Kap. 2.4.3), die ca. 225 Formate berücksichtigt (Safe 2009).

Für die Desktop Produkte der Testarchitektur war die Unterstützung der eingesetzten Datenbanksysteme und ihrer räumlichen Erweiterungen als Selektionskriterium minimal erforderlich. Die Interpretation der spezifischen Geometriedatentypen erfolgt bei diesen über integrierte Programmibliotheken oder -schnittstellen.

Die Software ArcGIS Desktop bietet darüber hinaus einen speziellen Interoperabilitätsadapter mit der Bezeichnung ArcGIS Data Interoperability Extension für den Zugriff auf Datenformate von ca. 70 verschiedenen Herstellern bzw. Quellen an. Diese Erweiterung basiert auf der Software FME und ermöglicht den direkten Zugriff, den einfachen Import-Export oder die komplexe Datentransformation (Esri 2008b). Im Rahmen dieser Testreihe wurde nur die Option des direkten Zugriffs ansatzweise untersucht. In ArcGIS Desktop lässt sich über diese Komponente eine direkte, nur lesende Verbindung zur Oracle oder PostgreSQL Datenbank aufbauen. Das Auslesen und die Visualisierung der Geodaten in den datenbankeigenen Speicherformaten ist in ArcGIS Desktop problemlos möglich. Die Interoperabilitätssoftware übernimmt dabei die Interpretation der räumlichen Datentypen.

Die Interpretation der Speicherformate oder Datentypen für Geodaten kann neben den integrierten Programmibliotheken gleichermaßen auch über spezielle Interoperabilitätsadapter erfolgen, sofern diese als Erweiterung für das Desktop System angeboten werden.

4.2.3 Datenbankverbindungen und Programmierschnittstellen

Für die allgemeine Informationsintegration stehen aus technischer Sicht unterschiedliche Arten von Infrastrukturkonzepten zur Verfügung (vgl. auch Leser & Naumann 2007). Im Rahmen dieser Arbeit liegt der Fokus auf der interoperablen Nutzung von Geodaten über den direkten Datenbankzugriff, die auch über eine virtuelle Integration der Geodaten, bei der keine zusätzlichen Zwischenschichten zur strukturellen Modifikation oder eine erneute

Materialisierung der Daten stattfinden, erreicht werden könnte. Zu Abschluss der Untersuchungsreihe werden die Ansätze der verteilten Datenbanken (VDB) und der programmiertechnischen Datenanbindung als Basis für weitere Nutzungskonzepte ventiliert.

Verbindung von Datenbanken

Eine nahe liegende Infrastruktur zur Integration von Daten stellen nach Leser und Naumann (2007) verteilte Datenbanken (VDB) dar. Zur Realisierung einer solchen Integrationsarchitektur können vier Ausprägungen unterschieden werden:

- Homogene verteilte Datenbanken basieren auf dem DBMS eines Herstellers und sind physisch verteilte Instanzen, die oftmals über proprietäre Protokolle kommunizieren.
- Heterogene VDB bestehen aus einem DBMS, das über produktspezifische Gateways auf Datenbanken anderer Hersteller zugreift.
- Heterogen generische VDB greifen über generische Schnittstellen auf Datenbanken anderer Hersteller zu.
- Der Zugriff auf nicht relationale Datenquellen aus DBMS über Wrapper.

Im Rahmen dieser Untersuchung wurden homogene und heterogen generische VDB zur virtuellen Integration von räumlichen Daten betrachtet. Im homogenen Architekturkonzept können sogenannte „Database Links“ als Verbindung zwischen verschiedenen verteilten Datenbanken eingerichtet werden, über die die Daten mittels entsprechenden Anfragen aus den jeweils anderen Instanzen abgerufen werden können. Sowohl das DBS Oracle als auch PostgreSQL unterstützen solche produktinternen Verbindungen. Über diese Technologie ist es grundsätzlich möglich, Geodaten, die in verteilten Datenbank eines Herstellers erstellt wurden, mittels einem Database Link zu integrieren und über einen GIS-Client auszulesen, was in der Testumgebung verifiziert werden konnte. Dieser Vorgang erfolgt ohne Mate-

realisierung der Daten in der zugreifenden Datenbank. Der Zugriff kann allerdings dadurch beschränkt werden, wenn die Daten von unterschiedlichen Erfassungssystemen generiert wurden und diese unterschiedliche Verwendungen des jeweiligen räumlichen Datentyps ermöglichen (z. B. Interpretationsprobleme, geometrische Fehler oder systemspezifische Restriktionen).

Sofern es sich nicht um Datenbanksysteme des gleichen Herstellers handelt und keine produktspezifischen Gateways existieren, bieten sich auch generische Schnittstellen wie ODBC oder JDBC für den Zugriff auf andere Datenbanken an. Ihre Verwendung ist aber mit einer Reihe von Nachteilen verbunden, wie beispielsweise die Anpassung der Syntax der Anfrage auf das Zielsystem. Für Geodaten stellt sich des Weiteren die Problematik, dass die geometrischen Attribute in den jeweiligen nutzerdefinierten, räumlichen Datentypen der Datenbanksysteme abliegen und somit dem jeweilig anderen DBMS nicht bekannt sind. Im Testfall wurde ein Database Link von Oracle nach PostgreSQL über ODBC aufgebaut. Von dem DBMS Oracle kann auf die Geodaten samt geometrischem Attribut, das als PG_Geometry in der PostgreSQL Tabelle abliegt, zugegriffen werden. Beispiel:

```
SELECT "id", "geom", "attribut1", "attribut2" FROM  
"geotabelle"@dblink2postgresql;
```

Die Notation @dblink2postgresql gibt an, in welcher angebundene Datenbank die Tabelle zu finden ist. Dieser Database Link muss zuvor im Oracle System definiert werden, wozu die Verbindungsparameter und ein berechtigter Benutzer auf die PostgreSQL Datenbank benötigt werden.

Eine in PostgreSQL erstellte Sicht auf eine Tabelle, die die Geometrien als PG_Geometry, WKT, WKB und GML ausgibt, kann im Oracle SQL Client problemlos angezeigt werden. Um ein korrektes Lesen zu ermöglichen, mussten aber alle Datentypen einer Typumwandlung zum Typ varchar unterzogen werden. Beispiel:

```
-- Erstellen einer Sicht in PostgreSQL
CREATE VIEW u01.geotab_v AS SELECT id, geom AS PGGEOM,
CAST(ST_AsText(geom) AS varchar) As WKT, CAST(ST_AsBinary(geom,'NDR')
AS varchar) As WKB, CAST(ST_AsGML(2, geom, 15) AS varchar) As GML2,
CAST(ST_AsGML(3, geom, 15) AS varchar) As GML3 FROM u01.geotab;

-- Abfrage aus der Oracle Datenbank
SELECT "id", "pggeom", "wkt", "wkb", "gml2", "gml3" FROM "geotab_v"@dblink;
-- Hier muss eine Anpassung der Syntax erfolgen, in dem die Spaltennamen in
Anführungszeichen eingefasst werden.

-- Ergebnisanzeige der Abfrage im Oracle SQL Client
1
0101000020EB7A000000000000F0B34A410000000070995441
POINT(3500000 5400000)
\001\001\000\000\000\000\000\000\000\000\360\263JA\000\000\000\000p\231TA
<gml:Point srsName="EPSG:31467"><gml:coordinates>3500000,5400000
</gml:coordinates></gml:Point>
<gml:Point srsName="EPSG:31467"><gml:pos>3500000 5400000</gml:pos>
</gml:Point>
```

Diese verschiedenen Ausgabeformen der Geometrien können nun in Oracle weiterverwendet werden, um beispielsweise Oracle-eigene SDO_Geometry Objekte zu erzeugen. Das gewünschte Ergebnis, d. h. die Verbindung von heterogenen verteilten Datenbanken zur virtuellen Integration von Geodaten, scheitert jedoch daran, dass die Anzeige der Geometrien in den Desktop GIS weder in den standardisierten Repräsentationsformen noch als Oracle SDO_Geometry (fehlender Spatial Index von nicht materialisierte Daten) möglich ist (vgl. Kap. 4.2.2 Repräsentation durch WKT, WKB und GML). Das Beziehen der Daten über die Datenbankverbindung und eine anschließende Materialisierung implizit Formatkonvertierung ist durchführbar, was allerdings einem ETL Prozess bzw. einer Kopie der Daten, aber nicht einer virtuellen Integration entspricht.

Die Integration von Daten über Architekturkonzepte verteilter Datenban-

ken ist in homogenen Softwareumgebungen auch für Geodaten realisierbar. In heterogenen VDB Landschaften ist die Durchführbarkeit einer virtuellen Datenintegration auf Datenbankebene softwareabhängig. Eine systemunabhängige Datenkombination in den Desktop Geoinformationssystemen konnte im Rahmen dieser Untersuchung nicht eruiert werden.

Programmiertechnische Zugriffsschnittstellen (API)

Die DBMS Oracle und PostgreSQL unterstützen diverse Datenbankzugriffsschnittstellen, die den Zugriff über herkömmliche Programmiersprachen wie beispielsweise C++ oder Java ermöglichen. Über diese Schnittstellen können nun vielfältige, programmiertechnische Anbindungen zwischen GI-Anwendungen und Datenbanken vorgenommen werden. Hierbei kann es sich um Erweiterungen über APIs von GIS-Software oder auch von primär nicht GIS-bezogenen Systemen, die die räumlichen Daten aus den Datenbanksystemen nutzen bzw. integrieren wollen, handeln. In diesem Kontext spielen auch Web-Applikationen gegenwärtig eine bedeutende Rolle.

Die prinzipiellen Ansätze eines programmiertechnischen Zugriffs auf die in Datenbanksystemen gespeicherten Geodaten sollen als eine grundlegende Technologie für Infrastrukturkonzepte zur Informationsintegration an dieser Stelle sondiert werden. Der für Datenbankanbindungen oder -anwendungen vollständige Funktionalitätsumfang von Programmiersprachen kann auf unterschiedliche Weise erreicht werden (Saake et al. 2008):

- SQL innerhalb einer konventionellen Programmiersprache (z. B. Java)
- Middleware zur Abbildung zwischen den Datenkonzepten der Programmiersprache und den Datenbankobjekten (z. B. Hibernate)
- Prozedurales SQL (z. B. Oracle PL/SQL)

Für die Datenbanksysteme der Testarchitektur wurde der erste Ansatz mittels der Programmiersprache Java (1.5) über die JDBC Schnittstelle durch-

geführt. In das Java Programm müssen dazu die entsprechenden JDBC Treiber des DBMS eingebunden werden.

Bei einem Zugriff auf die Oracle Database über ein Select Statement im Java Code können die Geometriedaten im Oracle SDO_Geometry Datentyp nicht direkt verwendet werden. Der Oracle-eigene räumliche Datentyp lässt sich nicht unmittelbar auf einen Java Datentyp abbilden. Eine solche SQL Klasse muss in Java als strukturierter Datentyp aufgefasst werden. Seit der Version SQL3 von 1999 (SQL:1999) existiert dazu der Datentyp STRUCT, der wiederum mit der JDBC Schnittstelle `java.sql.Struct` korrespondiert. Die Oracle-spezifische Klasse `oracle.sql.Struct` liefert hierzu das Gegenstück (Brinkhoff 2008). Somit kann nun ein Struct-Objekt für den SDO_Geometry Datentyp erstellt werden, um auf dessen einzelne Inhalte bzw. Felder (vgl. Typdefinition unter 4.2.1 Oracle Database) über ein Array zuzugreifen, wie folgendes Quellcode-Fragment verdeutlicht:

```
/* Beispiel für den Zugriff auf die Felder eines SDO_Geometry Punktobjektes */
Statement statement = dbverbindung.createStatement();
String sql = "SELECT geom FROM oracle_sdo_tab";
ResultSet ergebnis = statement.executeQuery(sql);
while (ergebnis.next()) {
    Struct geom = (Struct)ergebnis.getObject("geom");
    Object[] sdo_geom_inhalt = geom.getAttributes();
    // Attribut SDO_GTYPE
    BigDecimal gtype = (BigDecimal)sdo_geom_inhalt[0];
    // Attribut SDO_SRID
    BigDecimal srid = (BigDecimal)sdo_geom_inhalt[1];
    // Koordinaten aus SDO_POINT Array
    Struct punkt = (Struct)sdo_geom_inhalt[2];
    Object[] punktkoordinaten = punkt.getAttributes();
    BigDecimal xkoord = (BigDecimal)punktkoordinaten[0];
    BigDecimal ykoord = (BigDecimal)punktkoordinaten[1];
}
```

Da der Zugriff auf ein solches SQL-Objekt etwas umständlich ist, stellt

Oracle eine entsprechende Klassenbibliothek `oracle.spatial` zur Verfügung, die den Zugriff auf `SDO_Geometry` Objekte und deren Verwendung unterstützt bzw. vereinfacht (`sdoapi.jar`). Das Paket `oracle.spatial.geometry` enthält eine Klasse `JGeometry`, die Methoden zum Konstruieren, Abfragen, Konvertieren und zur Handhabung von Geometrieobjekten beinhaltet. Die Verwendung des Pakets `oracle.spatial` illustriert folgender Code-Auszug:

```
/* Beispiel für den Zugriff auf SDO_Geometry Punktobjekte unter Verwendung
der Klassenbibliothek oracle.spatial.geometry.JGeometry */
Statement statement = dbverbindung.createStatement();
String sql = "SELECT geom FROM oracle_sdo_tab";
ResultSet ergebnis = statement.executeQuery(sql);
while (ergebnis.next()) {
    STRUCT struct = (STRUCT) ergebnis.getObject("geom");
    JGeometry jGeom = JGeometry.load(struct);
    // Geometrietyp über Methode getType()
    int geomtyp = jGeom.getType();
    // Raumbezugssystem (SRID) über Methode getSRID()
    int srid = jGeom.getSRID();
    // Koordinaten des Punktobjektes
    double[] punktkoordinaten = jGeom.getPoint();
    double xkoord = punktkoordinaten[0];
    double ykoord = punktkoordinaten[1];
}
```

Für das DBMS PostgreSQL mit PostGIS als räumlicher Erweiterung kann ein vergleichbarer Zugriff realisiert werden. Die Geometrien in `PG_Geometry` können via SQL als Java Zeichenkette (String) oder über die Java Klasse `PGgeometry` des Pakets `org.postgis` (`postgis.jar`), die Methoden zur Konstruktion, Abfrage und Verwendung von PostGIS Geometrieobjekten bereitstellt, bezogen werden. Sofern die PostGIS Geometrie einem String zugewiesen wird, erhält man die kanonische, hexadezimale PostGIS EWKB Repräsentation. Geometriedaten als `PG_Geometry` Objekt können über Methoden ähnlich der `oracle.spatial` Bibliothek verarbeitet werden. Folgender Auszug

aus dem Quellcode veranschaulicht die Programmierung:

```
/* Beispiel für den Zugriff auf ein PG_Geometry Punktobjekt */
Statement statement = dbverbindung.createStatement();
ResultSet ergebnis = statement.executeQuery("SELECT geom FROM
pg01.postgre_pg_tab");
while( ergebnis.next() ) {
    // PG_Geometry als String
    String geom = ergebnis.getString(1);
    // Ausgabe z. B.:
    // 0101000020EB7A00000000000000DB4A4100000000F8AC5441
    // PG_Geometry als Object
    PGgeometry pggeom = (Pggeometry)ergebnis.getObject(1);
    // Ausgabe z. B.: SRID=31467;POINT(3520000 5420000)
    // Geometrietyp über Methode getGeoType()
    int gtype = pggeom.getGeoType();
    // Koordinaten des Punktobjektes
    Point punkt = (Point)pggeom.getGeometry();
    double xkoord = punkt.getX();
    double ykoord = punkt.getY();
    // Raumbezugssystem über Methode getSrid()
    int srid = punkt.getSrid(); }
```

- *Der Zugriff auf die Datenbanken und die darin abgelegten Geobjekte ist über entsprechende Schnittstellen in diverse Programmiersprachen generell möglich. Neben der Abfrage (Select) können auch Datenmanipulationen (Insert, Update, Delete) vorgenommen werden.*
- *Die Interpretation der nativen Geometriedatentypen bzw. ihrer Ausgabeformen obliegt dabei dem jeweiligen Anwendungsprogramm bzw. dem Datennutzer. Dies bietet eine maximale Flexibilität bzgl. der geometrischen Inhalte, aber auch eine große Komplexität, da die Auslegung der räumlichen Datentypen individuell erfolgen muss und nicht über eine entsprechend funktionale GIS-Software abgedeckt wird.*

5 Ergebnisse und Interpretation

In diesem Kapitel werden die Ergebnisse sowie deren Interpretation und Beurteilung hinsichtlich der Fragestellungen dieser Arbeit, aus der Synthese der vorausgehenden theoretischen Grundlagen und der praktischen Empirie, dokumentiert und daraus Schlussfolgerungen abgeleitet.

5.1 Modelle und Technologien

Fragestellung:

- *Wie lassen sich die existierenden Modelle und Technologien zur technischen Realisierung des direkten Zugriffs auf Geodaten in Datenbanksystemen einordnen und wie tragen sie zu dieser bei?*
- *Welche technischen Voraussetzungen müssten in heterogenen Systemlandschaften seitens der Geodatenbank- und Geoinformationssysteme erfüllt sein, um einen direkten Zugriff zur interoperablen Geodatennutzung zu ermöglichen? Welche Einschränkungen sind zu berücksichtigen?*

Im Kontext der Abbildung von geometrischen Eigenschaften realweltlicher Objekte in Datenbanksystemen sind besonders die in Kapitel 2 vorgestellten Normen ISO/IEC SQL/MM Spatial und ISO Simple Feature Access von Bedeutung. Sie definieren die Modellierung von Geometrien sowie deren Konstruktion, den Zugriff und die Abfrage von topologischen Beziehungen via SQL Methoden. Als Konstruktionsvorschrift dient hier vor allem die Repräsentationsform Well-known Text. Die Speicherung der Geometrien kann in vordefinierten (numerische oder binäre, nur gültig für SFA) oder nutzerdefinierten Datentypen erfolgen. Die Struktur der nutzerdefinierten Geometriedatentypen wird durch die Normen aber nicht beschrieben.

Der Softwaremarkt bietet gegenwärtig objektrelationale DBMS oder auch GIS-spezifische Datenbankmiddleware, die diese relevanten Normen auf-

greifen und als räumliche Erweiterungen, zumindest partiell, implementieren. Dadurch ist es möglich, alle grundlegenden, geometrisch einfachen Geoobjekte gemäß den genormten Geometrieklassen zu modellieren und sie somit hinsichtlich ihrer geometrischen Form auch über Systemgrenzen hinweg interoperabel zu nutzen und auszutauschen.

Aus technischer Sicht stehen für die Datenbankverbindung ausgereifte, dem ISO SQL/CLI Standard folgende oder systemspezifische Zugriffsschnittstellen zu den DBMS bereit. Die Kommunikation zwischen Datenbank und Anwendung wird über die standardisierte Abfragesprache SQL und mit ihrer Erweiterung SQL/MM Spatial durchgeführt. Aufgrund der systemspezifischen Implementierungen der räumlichen Datenbankerweiterungen gestaltet sich die Erstellung von Geometrien, der Zugriff und die Analyse via SQL Methoden zumindest partiell sehr divergent. Die Speicherung der Geometrien erfolgt in den berücksichtigten Systemen über räumliche Datentypen, deren Struktur herstellerspezifisch ist.

Dies hat nun zur Folge, dass zwar die standardisierte Modellierung und die Abfrage von Geodaten in den Datenbanksystemen via SQL grundsätzlich möglich sind, die Interpretation der systemspezifischen Ausgabeformate der geometrischen Attribute aber immer der zugreifenden Anwendung obliegt.

Die Simple Feature Geometrieklassen konnten von den getesteten Desktop Systemen unter Verwendung der jeweils datenbankspezifischen Speicherformen weitestgehend interpretiert bzw. visualisiert werden und können somit als allgemeine („kleinste“) Interoperabilitätsebene eingesetzt werden. Eine Ausnahme stellen hier heterogene GeometryCollections dar, deren Interpretation softwareabhängig ist. Eine Zerlegung in die einzelnen Geometrieklassen über Datenbanksichten ermöglicht jedoch meist die Anzeige. Hinsichtlich nicht einfacher Geometrien und spezieller Ausprägungen zeichnet sich zur Zeit noch ein anderes Bild. Die Modellierung von Kreisbögen

konnte in die Datenbanksystemen der Testumgebung zwar durchgeführt werden, die korrekte Interpretation durch die eingesetzten Desktop GIS ist allerdings sehr heterogen bzw. softwareabhängig. Dahingegen sind linienhafte Verbindungen aus eleganten Kurven (Splines), wie sie in der Norm ISO Spatial Schema vorgestellt werden, weder in den datenbankrelevanten Standards SQL/MM Spatial oder SFA noch in den Datenbanksystemen zu finden.

Ein weiteres Interoperabilitätsdefizit besteht noch bei räumlich platzierten Beschriftungen, für die keine oder nur proprietäre Umsetzungen zu finden sind (OGC 2006a). Die OGC SFA 1.2.0 offeriert hier ein Modell, das in der aktuellen Version der Oracle Database 11g im Gegensatz zur Kombination PostgreSQL mit PostGIS zwar aufgenommen, gegenwärtig aber noch nicht umfassend dokumentiert und fortgeschritten ist.

Es lässt sich resultieren, dass die standardisierten Modelle und Technologien für die Abbildung von geometrisch einfachen Geodaten in Datenbanksystemen und den Zugriff auf diese sowohl ihre Verbreitung in den Softwareprodukten gefunden haben, als auch eine entscheidende Rolle für die interoperable Nutzung und die Integration von Geodaten in heterogenen Architekturen auf Basis von Geodatenbanksystemen spielen. Die Interoperabilität beschränkt sich dabei aber auf die geometrische Modellierung und die Zugriffs- und Analysemethoden im SQL Sprachumfang. Die unterschiedlichen Realisierungen der räumlichen Datenbankerweiterungen müssen durch Geoinformationssysteme oder andere Anwendungen für den direkten Zugriff auf die Geometrien via SQL individuell berücksichtigt bzw. unterstützt werden, d. h. entsprechende Anpassungen müssen vorgenommen und ggf. Interpreter enthalten sein. Eine vollständige Konformität zu den Normen SQL/MM Spatial oder SFA garantiert somit eine standardisierte Modellierung und Abfrage via SQL, aber nicht die Lesbarkeit der Geometrien, sofern diese in nutzerdefinierten und somit systemspezifischen Geometrie-

datentypen abgelegt und ausgelesen werden, da deren Definition durch die Normen nicht festgelegt ist.

Eine interoperable Geodatennutzung in einer heterogenen Architektur über direkte Datenbankzugriffe setzt die Interpretation von spezifischen, nicht genormte Speicherformen bzw. deren Ausgabeformaten voraus, sofern nicht die standardisierten Repräsentationsformen (WKT, WKB oder GML) genutzt werden. Diese Austauschformate konnten in den Desktop GIS der Testumgebung im Regelfall nicht direkt verwendet respektive nicht als Ausgabeform angegeben werden.

Die Abbildung von realweltlich komplexen, geometrischen oder topologischen Modellen sowie die georeferenzierte Objektbeschriftung bedarf einem weiteren Fortschreiten der relevanten Standards und deren Implementierung.

5.2 Anforderungen und Einsatzbereiche

Fragestellung: Für welche Anforderungen und Einsatzbereiche sind direkte Zugriffe auf Geodatenbanksysteme in heterogenen Architekturen zur interoperablen Geodatennutzung geeignet?

Ein direkter Datenbankzugriff, wie er für diese Arbeit im Kapitel 3 definiert wurde, zeichnet sich vor allem durch eine maximale Flexibilität für den Anwender bzw. Datennutzer aus. Sofern die technischen Voraussetzungen das Lesen der Geodaten mit Geometrie- und Sachattributen durch das Anwendungssystem ermöglichen, verfügt der Datennutzer im Rahmen seiner Autorisierung durch den Datenbereitsteller über den vollen bzw. benötigten Datenumfang oder das Datenbankschema und kann diesen für seine individuellen Anforderungen konfigurieren und in seinen Anwendungssystemen kombinieren. Der Datennutzer hat beim direkten Datenbankzugriff auch unmittelbar Zugang zu den vektorialen Daten, die eine erweiterte Datenvisualisierung und -auswertung ermöglichen. Die Anzeige von Geometrie- und

zugehörigen Sachdaten, die Kombination von Objektklassen, auch aus unterschiedlichen Datenbanksystemen, sowie die Erstellung von Datenbanksichten auf die Geodatentabellen konnte durch die Testumgebung verifiziert werden. Der Datenbereitsteller kann wiederum die vorhandenen und bekannten Benutzerverwaltungskonzepte und Sicherheitsmechanismen des DBMS zur Kontrolle und Freigabe der Datenbestände nutzen.

Diese maximale Datenflexibilität und Nutzungsfreiheit ist allerdings mit einer erhöhten Zugriffskomplexität verbunden, d. h. sie setzt das Wissen über Datenstruktur und -modell, über die Konfiguration innerhalb der Anwendung sowie ggf. über Visualisierungseigenschaften auf Seiten des Datennutzers voraus. Der Anwender erhält quasi die Rohdaten und keine durch den Datenbereitsteller aufbereitete Datenansicht (Karte).

Des Weiteren bietet sich auch für primär Nicht-GIS-Anwendungen über diverse Datenbankzugriffsschnittstellen (APIs) die Möglichkeit der Geodatenintegration. Durch die SQL Methoden der räumlichen Datenbankeerweiterungen können Geodaten unabhängig von einem Geoinformationssystem erstellt, manipuliert und gelesen werden. Die partiell sehr divergierenden, systemspezifischen Realisierungen erschweren allerdings eine einheitliche Verwendung. Häufig reichen in Nicht-GIS-Anwendungen auch die Informationen über topologischen Beziehungen zwischen Geobjekten aus, die mit räumlichen Funktionen analysiert werden können. Durch die Standardisierungen im Bereich der Geodatenbanksysteme hinsichtlich der Modellierung von Geobjekten und dem Zugriff auf diese mittels SQL steht folglich jedem Client, der eine Datenbankverbindung aufbauen kann, die Option offen, zumindest die Sachattribute der Geobjekte, die standardisierten Repräsentationsformen der Geometrien und über die SQL Methoden die topologischen Beziehungen abzufragen, auch wenn die systemspezifischen Ausgabeformate der Geometrien durch ihn nicht interpretiert werden können.

Ein offene Geodatenhaltung in Datenbanksystemen mit einer standardisier-

ten Modellierung und SQL Zugriffsmethoden erleichtert die Integration in die herkömmliche, unternehmensweite IT-Landschaft und erhöht die Interoperabilität bezüglich der Geodatennutzung.

5.3 Integration in IT- und GIS-Architekturen

Fragestellung: Wie kann die Integration dieses Ansatzes des direkten Zugriffs in eine organisationsweite IT- und GIS-Strategie respektive -Landschaft, vornehmlich auch im Hinblick auf serviceorientierte Architekturen, aussehen?

Der direkter Zugriff auf die Datenhaltung kommt innerhalb eines abgeschlossenen Einzelsystems (Quell- oder Fachsystem) zur Datenerfassung und -pflege meist zwangsläufig zum Einsatz. Viele IT-Anwendungen und im Speziellen auch GIS setzen heute als Datenhaltungskomponente Datenbanksysteme ein. Die Anbindung eines oder mehrerer verschiedener Geodatenbanksystemen ist in den GI-Systemen somit häufig bereits realisiert. In heterogenen GIS-Landschaften mit unterschiedlichen Desktop- oder Web-Geoinformationssystemen implizit verteilten Geodatenbanken kann der Ansatz des direkten Zugriffs folglich ein Architekturkonzept, wie beispielsweise eine serviceorientierte Architektur auf Basis von standardisierten OGC Web Services zur interoperablen Geodatennutzung, auf den im Folgenden aufgrund seiner weiten und zunehmenden Verbreitung vornehmlich Bezug genommen wird, ergänzen, steht dazu aber nicht unmittelbar in Konkurrenz. Diese Ergänzung bietet sich bei bestimmten Anforderungen oder in speziellen Anwendungsfällen an, in denen der direkte Zugriff technisch möglich ist und seine Vorteile gegenüber den webbasierten Geodiensten überwiegen. Dies kann je nach technischen und strukturellen Voraussetzungen dann der Fall sein, wenn

- eine maximale Flexibilität hinsichtlich der Konfiguration und Abfrage der Geodaten benötigt wird und dazu kein vordefinierter, ggf. unverän-

derbarer, Service eingesetzt werden kann

- eine sehr hohe Datenaktualität gefordert wird
- durch den direkten Zugriff ohne Zwischenschichten eine erhöhte Performance erreicht werden kann und diese für die Anwendung notwendig ist
- die Geoinformationssystemsoftware den direkten Datenbankzugriff und den systemspezifischen Geometriedatentyp besser unterstützt als beispielsweise eine OGC Web-Schnittstelle oder der Applikationsserver, der diese bereitstellt; dies kann in der Praxis dadurch auftreten, da GI-Systeme zunehmend Geodatenbanksysteme als primäre Datenhaltungskomponente einsetzen und somit die Datenbankanbindung zwangsläufig implementiert haben
- die finanziellen Aufwände für die Anschaffung, Administration, Wartung und Know-how einer Datenbereitstellung über beispielsweise OWS größer als bei einem direkten Datenbankzugriff sind; diese wirtschaftliche Erwägung könnte gerade bei großen Unternehmen mit verschiedenen, autonomen Organisationseinheiten, die ihre Daten zur Verfügung stellen wollen, auftreten; hier können aber auch andere Konzepte, wie beispielsweise eine zentrale, unternehmensweite Geodatenbank als Geo-Data-Warehouse, eingesetzt werden
- der Zugriff auf komplexe Geometrien oder topologische Modelle notwendig ist und dieser nur über eine systemspezifische Implementierung möglich ist bzw. nur über eine direkte Anbindung des Geodatenbanksystems erfolgen kann
- ein individueller Zugriff oder eine Anbindung von Nicht-GIS-Anwendungen via SQL realisiert werden soll

Demgegenüber stehen aber auch einige, zum Teil bereits erwähnte Nachteile gegenüber den standardisierten OGC Web Services, wie

- die erhöhte Zugriffskomplexität auf Seiten des Datennutzers
- keine vordefinierten, spezifizierten Dienstleistungen, wie beispielsweise vorgenerierte Karten eines WMS, die auf einfache Weise eingebunden und genutzt werden können; diese könnten gerade auch dann Verwendung finden, wenn komplexe Geometrien, Modelle oder auch Beschriftungen vereinfacht ausgetauscht werden sollen
- technische Inkompatibilität und mangelnde Interoperabilität aufgrund fehlender Interpretationskomponenten der systemspezifischen Ausgabeformate der Geometriedatentypen, Versionsunterschieden der Software oder der implementierten Standards sowie technische oder organisatorische Verbindungsbarrieren zur Datenbank

Ein direkter Zugriff auf verteilte Geodatenbanken ist nicht vorteilhaft oder eignet sich generell nicht, wenn eine Notwendigkeit zur Transformation, Umstrukturierung oder Bereinigung der Daten besteht. Hier ist meist eine Materialisierung der Daten im Zielsystem erforderlich. Häufig findet hier das Konzept des Datentransfers mit seinen bekannten Vor- und Nachteilen seinen Einsatz, beispielsweise auch in Architekturen, die auf einer zentralen, unternehmensweiten Geodatenbank aufbauen. Diese zentrale Geodatenhaltung kann nun sowohl zwei- oder mehrschichtige Zugriffsmöglichkeiten für die verschiedenen Clients anbieten.

Eine praktikable, virtuelle Integration von verteilten, heterogenen Geodatenbanken über die Verbindung von Datenbanksystemen als Erweiterung des Ansatzes konnte über generische Schnittstellen nicht eruiert werden.

6 Zusammenfassung und Ausblick

Das letzte Kapitel gibt einen zusammenfassenden Rückblick sowie einen allgemeinen Ausblick hinsichtlich Zielsetzung, Lösungsansatz, Durchführung und Ergebnis dieser Master Thesis.

6.1 Zusammenfassung

In dieser Master Thesis sollten Erkenntnisse zur technischen Qualifizierung und dem strategischen Einsatz von direkten Zugriffen zu geographischen Datenbanksystemen für die interoperablen Geodatennutzung in heterogenen GIS-Landschaften gewonnen werden.

Dazu wurde zunächst eine eingehende Betrachtung des aktuellen Entwicklungsstandes auf diesem Gebiet sowie die Herleitung von Begriffsdefinitionen und einer Hypothese als theoretischer Ausgangspunkt der Untersuchung durchgeführt. Ein direkter Datenbankzugriff definiert sich demnach innerhalb dieser Arbeit als ein Zugriff eines Geoinformations- oder sonstigen Clients auf die Geodaten in einem Geodatenbanksystem via SQL, ohne eine zusätzliche Zwischenschicht, die die Daten strukturell modifiziert, transformiert oder materialisiert. Die Annahme bestand nun darin, dass diese direkten Datenbankzugriffe unter bestimmten Voraussetzungen und für spezielle Anforderungen, vor allem innerhalb einer Organisation, als ein adäquates Verfahren für die interoperable Geodatennutzung in heterogenen (GI-)Softwarearchitekturen eingesetzt werden können. Im Einzelnen waren dazu die existierenden Modelle und Technologien, die technischen Voraussetzungen sowie die Integrationsmöglichkeiten des Ansatzes in eine IT- und GIS-Landschaft zu untersuchen. Die methodische Vorgehensweise bestand darin, über eine technisch-orientierten Empirie die theoretische Annahme zu evaluieren und daraus verallgemeinerte Aussagen der wissenschaftli-

chen Untersuchung abzuleiten.

Die praktische Durchführung erfolgte über eine exemplarische Testumgebung auf Basis heterogener Geodatenbank- und Geoinformationssysteme anhand derer spezifizierte Testfälle als Indikatoren durchlaufen wurden. Die Einzelereignisse der Testfälle wurden in einer Synthese zusammengeführt, bewertet, interpretiert und lieferten die resultierenden Aussagen der Untersuchung.

Als Ergebnis dieser wissenschaftlichen Arbeit kann zusammenfassend festgehalten werden, dass die relevanten Standards die interoperable Nutzung von verteilten Geodatenbanken hinsichtlich der Datenmodellierung sowie dem Zugriff und der Analyse via SQL sicherstellen, und diese, wenn auch noch unvollständig und partiell sehr divergent, ihre Verbreitung in den Softwareprodukten gefunden haben. Die unterschiedlichen Realisierungen der räumlichen Datenbankeerweiterungen, hier im Besonderen die systemspezifischen Geometriedatentypen, deren Definition durch die Standards nicht festgelegt ist, erfordern aber individuelle Anpassungen auf das DBMS durch die Clients, was eine vollständige Interoperabilität beschränkt. Hinsichtlich des Einsatzbereiches gewährt ein direkter Datenbankzugriff vor allem eine maximale Flexibilität für den Datennutzer bzw. die datennutzende Anwendung, die allerdings mit einer erhöhten Komplexität der Datenintegration, -konfiguration und -repräsentation einhergeht. In heterogenen GIS-Landschaften kann der Ansatz des direkten Datenbankzugriffs folglich Architekturkonzepte, wie beispielsweise die zunehmend verbreitete, serviceorientierte Architektur auf Basis von standardisierten OGC Web Services zur interoperablen Geodatennutzung, unter bestimmten Voraussetzungen und für spezielle Anforderungen ergänzen und dabei auch Vorteile gegenüber den webbasierten Geodiensten aufweisen.

6.2 Ausblick

Die vorgestellten Ergebnisse und deren Interpretation in Kapitel 5 beinhalten bereits eine Generalisierung der Resultate aus der Testumgebung mit dem Ziel der Übertragbarkeit auch auf andere Softwarekonstellationen. Inwieweit eine Verallgemeinerung in Anbetracht der Vielfalt an Software und Lösungen auf dem Markt der geoinformationsverarbeitenden Produkte allerdings zutreffend ist, kann an dieser Stelle nicht abschließend beantwortet werden. Aufgrund der gezielten Auswahl der Komponenten der Testarchitektur und den erhaltenen Ergebnissen kann aber davon ausgegangen werden, dass ähnliche Verhaltensweisen und Resultate auch in anderen Zusammensetzungen zu erwarten sind.

Die Auseinandersetzung und die Untersuchung der eingangs näher beschriebenen Thematik der interoperablen Geodatenutzung auf Basis von Geodatenbanksystemen respektive einer offenen Geodatenhaltung sowie der Zugriff auf verteilte, heterogene Geodatenbanken hat gezeigt, dass die Standardisierungsbemühungen in diesem Bereich nicht nur entscheidende Impulse geben, sondern vor allem auch Implementierungsspezifikationen hervorbringen, die auch ihren Einzug in die Softwareprodukte gefunden haben. Dieser Zustand trägt insgesamt zu einer offenen Geodatenhaltung und gesteigerter Interoperabilität bei. Die Implementierungsfreiheiten und die herstellerspezifischen, zum Teil weiterführenden Realisierungen und Erweiterungen in den Datenbanksystemen haben allerdings zur Folge, dass gegenwärtig systemspezifische Umsetzungen immer zu berücksichtigen sind. Eine ganzheitliche Interoperabilität hinsichtlich eines interoperablen, direkten Zugriffs auf räumliche Vektordaten in Datenbanksystemen könnte über die Entwicklung eines standardisierten, räumlichen Datentyps bzw. einer Typstruktur oder einer erweiterten Nutzung der Repräsentationsformen verfolgt werden.

Literaturverzeichnis

(Brinkhoff 2006)

Brinkhoff, T., 2006. Geodatenbanksysteme: Gestern, Heute und Morgen. *Photogrammetrie, Fernerkundung, Geoinformation (PFG)*, Heft 5, S. 379-386.

(Brinkhoff 2007a)

Brinkhoff, T., 2007. Räumliche Netzwerk- und Topologiedatenbanken. *Datenbank-Spektrum*, Heft 21, S. 15-21.

(Brinkhoff 2007b)

Brinkhoff, T., 2007. Open-Source-Geodatenbanksysteme. *Datenbank-Spektrum*, Heft 22, S. 37-43.

(Brinkhoff 2008)

Brinkhoff, T., 2008. *Geodatenbanksysteme in Theorie und Praxis: Einführung in objektrelationale Geodatenbanken unter besonderer Berücksichtigung von Oracle Spatial*. 2. Auflage. Heidelberg: Herbert Wichmann Verlag.

(Datacom 2008)

DATAKOM Buchverlag GmbH, 2008. *IT Wissen: Das große Online-Lexikon für Informationstechnologie*. [Online]. DATAKOM Buchverlag GmbH. Erreichbar unter: <http://itwissen.info> [Zugriff am 23.11.2008]

(disy 2007)

disy Informationssysteme GmbH, 2007. *disy GISterm Version 3.9 Administration und Konfiguration*. Karlsruhe.

(Donaubauer 2004)

Donaubauer, A.J., 2004. *Interoperable Nutzung verteilter Geodatenbanken mittels standardisierter Geo Web Services*. Dissertation Dr., Technische Universität München.

(Dunkel et al. 2008)

Dunkel, J. et al., 2008. *Systemarchitekturen für verteilte Anwendungen: Client-Server, Multi-Tier, SOA, Event-Driven Architectures, P2P, Grid, Web 2.0*. München: Carl Hanser Verlag.

(Esri 2008a)

Environmental Systems Research Institute Inc. (ESRI), 2008. *ArcGIS Server Help*. Redlands.

(Esri 2008b)

Environmental Systems Research Institute Inc. (ESRI), 2008. *ArcGIS Desktop Help*. Redlands.

(Esri 2009)

Environmental Systems Research Institute Inc. (ESRI), 2009. ESRI. [Online]. Erreichbar unter: <http://www.esri.com> [Zugriff am 30.04.2009]

(GeoZG 2009)

Geodatenzugangsgesetz, 2009. *Gesetz über den Zugang zu digitalen Geodaten (Geodatenzugangsgesetz - GeoZG)*. Bundesrepublik Deutschland. [Online]. Erreichbar unter: <http://bundesrecht.juris.de/bundesrecht/geozg/gesamt~pdf> [Zugriff am 27.04.2009]

(Greener 2008)

Greener, S., 2008. *An Introduction to Spatial Databases*. The SpatialDB Ad-

visor. [Online]. Erreichbar unter: www.spatialdbadvisor.com/file_download/~35/BADSIG_November_11th.pdf [15.02.2009]

(Güting 1994)

Güting, R.H., 1994. An Introduction to Spatial Database Systems. *VLDB Journal*, Band 3, Nummer 4, S. 357-399.

(gvSig 2007)

Generalitat Valencia, 2007. gvSIG 1.1 User guide (English-Version 1). [Online]. Generalitat Valencia Conselleria de Infraestructuras y Transporte. Erreichbar unter: <http://www.gvsig.gva.es> [Zugriff am 24.07.2008]

(Hibernate Spatial 2008)

Hibernate Spatial, 2008. Hibernate Spatial. Geovise BVBA. [Online] Erreichbar unter <http://www.hibernatepatial.org> [Zugriff am 20.12.2008].

(INSPIRE 2007)

Europäische Union, 2007. Richtlinie 2007/2/EG des Europäischen Parlaments und des Rates vom 14. März 2007 zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE). [Online]. Europäische Union. Erreichbar unter: <http://eur-lex.europa.eu/LexUriServ/LexUri~Serv.do?uri=OJ:L:2007:108:0001:0014:DE:PDF> [Zugriff am 30.11.2008]

(ISO 2008)

International Organization for Standardization (ISO), 2008. *International Organization for Standardization. International Standards for Business, Government and Society*. [Online]. ISO. Erreichbar unter: <http://www.iso.org> [Zugriff am 30.11.2008]

(Kothuri et al. 2004)

Kothuri, R. & Godfrind, A. & Beinat, E., 2004. *Pro Oracle Spatial: The essential guide to developing spatially enabled business applications*. Berkeley: Apress.

(Leser & Naumann 2007)

Leser, U. & Naumann, F., 2007. *Informationsintegration: Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen*. Heidelberg: dpunkt.verlag GmbH.

(Microsoft 2008)

Microsoft Corporation, 2008. Microsoft® SQL Server® 2008 Spatial Data. Delivering location intelligence within your applications. [Online]. Microsoft. Erreichbar unter <http://www.microsoft.com/sqlserver/2008/en/us/spatial-data.aspx> [Zugriff am 07.12.2008]

(OGC 2001)

Open Geospatial Consortium, Inc.® (OGC), 2001. *OpenGIS® Implementation Specification: Coordinate Transformation Services*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/index.php?artifact_id=999 [Zugriff am 06.12.2008]

(OGC 2002)

Open Geospatial Consortium, Inc.® (OGC), 2002. *The OpenGIS Abstract Specification. Topic 12: OpenGIS Service Architecture. Version 4.3*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=12~21 [Zugriff am 30.10.2008]

(OGC 2004)

Open Geospatial Consortium, Inc.[®] (OGC), 2004. *OpenGIS[®] White Paper: Integrating Geospatial Standards and Standards Strategies into Business Process*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?~artifact_id=5098&version=2&format=pdf [Zugriff am 13.10.2008]

(OGC 2005a)

Open Geospatial Consortium, Inc.[®] (OGC), 2005. *OpenGIS[®] Interoperability & Open Architecture: An Analysis of Existing Standardisation Process & Procedures. Version 1.0*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=10594 [Zugriff am 14.10.2008]

(OGC 2005b)

Open Geospatial Consortium, Inc.[®] (OGC), 2005. *Web Feature Service Implementation Specification: Version 1.1.0*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=8339 [Zugriff 30.10.2008]

(OGC 2005c)

Open Geospatial Consortium, Inc.[®] (OGC), 2005. *OpenGIS[®] Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture: Version 1.1.0*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=13227 [Zugriff am 30.10.2008]

(OGC 2005d)

Open Geospatial Consortium, Inc.[®] (OGC), 2005. *OpenGIS[®] Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option: Version 1.1.0*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=13228 [Zugriff am 30.10.2008]

(OGC 2006a)

Open Geospatial Consortium, Inc.[®] (OGC), 2006. *OpenGIS[®] Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture: Version 1.2.0*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=18241 [Zugriff am 30.10.2008]

(OGC 2006b)

Open Geospatial Consortium, Inc.[®] (OGC), 2006. *OpenGIS[®] Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option: Version 1.2.0*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=18242 [Zugriff am 30.10.2008]

(OGC 2006c)

Open Geospatial Consortium, Inc.[®] (OGC), 2006. *OpenGIS[®] Web Map Server Implementation Specification: Version 1.3.0*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=14416 [Zugriff am 30.10.2008]

(OGC 2006d)

Open Geospatial Consortium, Inc.[®] (OGC), 2006. *OpenGIS[®] Geography Markup Language (GML) simple feature profile: Version 1.0*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=15201 [Zugriff am 29.12.2008]

(OGC 2007)

Open Geospatial Consortium, Inc.[®] (OGC), 2007. *OpenGIS[®] Geography Markup Language (GML) Encoding Standard: Version 3.2.1*. [Online]. OGC. Erreichbar unter: http://portal.opengeospatial.org/files/?artifact_id=20509 [Zugriff am 14.10.2008]

(OGC 2009)

Open Geospatial Consortium, Inc.[®] (OGC), 2009. *The Open Geospatial Consortium, Inc.[®] (OGC)*. [Online]. OGC. Erreichbar unter: <http://www.opengeo~spatial.org> [Zugriff am 17.03.2009]

(OGP 2008)

International Association of Oil & Gas Producers (OGP), 2008. OGP Surveying & Positioning Committee. [Online]. OGP. Erreichbar unter: <http://www.~epsg.org> [Zugriff am 20.12.2008]

(Oracle 2006)

Oracle Corporation, 2006. Oracle[®] Spatial User's Guide and Reference 10g Release 2 (10.2). B14255-03. [Online]. Oracle Corporation. Erreichbar unter: <http://www.oracle.com> [Zugriff am 12.09.2008]

(Oracle 2007)

Oracle Corporation, 2007. Oracle[®] Spatial Developer's Guide 11g Release 1 (11.1). B28400-02. [Online]. Oracle Corporation. Erreichbar unter: <http://~www.oracle.com> [Zugriff am 12.09.2008]

(PostgreSQL 2008)

PostgreSQL Global Development Group, 2008. *PostgreSQL. The world's most advanced open source database*. [Online]. PostgreSQL Global Development Group. Erreichbar unter: <http://www.postgresql.org> [Zugriff am 26.10.2008]

(PostGIS 2008a)

PostGIS, 2008. *PostGIS open source community site*. [Online]. Refrations Research. Erreichbar unter: <http://www.postgis.org> [Zugriff am 26.10.2008]

(PostGIS 2008b)

PostGIS, 2008. *PostGIS 1.3.5 Manual*. [Online]. Refrations Research. Erreichbar unter: <http://www.postgis.org/documentation/> [Zugriff 21.12. 2008]

(Saake et al. 2008)

Saake, G. & Sattler, K.-U. & Heuer, A., 2008. *Datenbanken: Konzepte und Sprachen*. 3. Auflage. Heidelberg: mitp Verlag.

(Safe 2009)

Safe Software Inc., 2009. *Safe Software Inc., Feature Manipulation Engine*. [Online]. Safe Software Inc.. Erreichbar unter: <http://www.safe.com> [Zugriff am 10.02.2009]

(Schanz 2006)

Schanz, C., 2006. *Freie Datenbanksysteme zur Verwaltung, Analyse und Bearbeitung von Geodaten. Räumliche Informationen in freien Datenbanken*. Master Thesis, Universität Salzburg.

(Scheugenpflug 2005)

Scheugenpflug, S.J., 2005. *Relationale und Objektrelationale Datenbankkonzepte in Geoinformationssystemen*. Dissertation Dr., Technische Universität München.

(Stolze 2006)

Stolze, K., 2006. *Integration of Spatial Vector Data in Enterprise Relational Database Environments*. Dissertation Dr., Universität Jena.

(Yeung & Hall 2007)

Yeung, A.K.W. & Hall, G.B., 2007. *Spatial Database Systems: Design, Implementation and Project Management*. Dordrecht: Springer

Anhang

Räumliche Datentypen - Definition, Konstruktion und Zugriff

Tab. 1: Räumliche Datentypen / Speicherformate

Räumlicher Datentyp/ Speicherformat	Definition des räumlichen Datentyps	Beispiele für die Standardausgabe eines Punktoobjekts (DHDN GK 3, 3500000, 54000000) bei einer SQL-Abfrage
Oracle SDO_GEOMETRY (Oracle 2007)	MDSYS.SDO_GEOMETRY(SDO_GTYPE number, SDO_SRID number, SDO_POINT sdo_point_type, SDO_ELEM_INFO sdo_elem_info_array, SDO_ORDINATES sdo_ordinate_array)	SDO_GEOMETRY(2001, 31467, SDO_POINT_TYPE(35000000, 5400000, NULL), NULL, NULL)
Oracle ST_GEOMETRY (Oracle 2007)	MDSYS.ST_GEOMETRY(GEOM sdo_geometry)	ST_GEOMETRY(SDO_GEOMETRY(2001, 31467, SDO_POINT_TYPE(3500000, 5400000, NULL), NULL, NULL)) oder bei Konstruktion mit ST_POINT ST_POINT(SDO_GEOMETRY(2001, 31467, SDO_POINT_TYPE(3500000, 5400000, NULL), NULL, NULL))
PostgreSQL/PostGIS PG_GEOMETRY (PostGIS 2008b)	Erweitertes WKT oder WKB (PostGIS EWKT bzw. EWKB erweitert um SRID und 3dm, 3dz, 4d). Beispiel EWKT: 'SRID=31467;POINT(3500000 5400000)'	0101000020EB7A00000000000000F0B34A410000000070995441
ESRI ST_GEOMETRY in Oracle (Esri 2008a)	SDE.ST_GEOMETRY (ENTITY integer, NUMPTS integer, MINX float(64), MINY float(64), MAXX float(64), MAXY float(64), MINZ float (64), MAXZ float(64), MINM float(64), MAXM float(64), AREA float(64), LEN float(64), SRID integer, POINTS blob)	ST_GEOMETRY(1, 1, 3510000, 5440000, 3510000, 5440000, NULL, NULL, NULL, NULL, 0, 0, 2, '0C000000001000000080ADDAD3A30380FAECB8FE08') oder bei Konstruktion mit ST_POINT ST_POINT(1, 1, 3510000, 5440000, 3510000, 5440000, NULL, NULL, NULL, NULL, 0, 0, 2, '0C000000001000000080ADDAD3A30380FAECB8FE08')
ESRI ST_GEOMETRY in PostgreSQL (Esri 2008a)	SDE.ST_GEOMETRY (SIZE long integer, SRID long integer, NUMPTS long integer, ENTITY short integer, SQLTYPE short integer, MINX lfloat, MINY lfloat, MAXX lfloat, MAXY lfloat, MINZ lfloat, MAXZ lfloat, MINM lfloat, MAXM lfloat, AREA lfloat, LEN lfloat, SHAPE bytea)	2800000001000000001000400200000000C00000001000000 ~ 80A9ABF4A20380EAB0BBFB0800000000

*Tab. 2: Konstruktoren und Zugriffsmethoden
(Anm.: Auszug, vollständige Dokumentation siehe Benutzerhandbücher und Software)*

Räumlicher Datentyp/ Speicherformat	Konstruktoren	Zugriffsmethoden
Oracle SDO_GEOMETRY (Oracle 2007)	-- Oracle SDO_Geometry SDO_GEOMETRY(SDO_GTYPE, SDO_SRID, SDO_POINT, SDO_ELEM_INFO, SDO_ORDINATES) -- WKT und WKB SDO_GEOMETRY(wkt, srid) SDO_UTIL.FROM_WKTGEOMETRY(wkt) SDO_GEOMETRY(wkb, srid) SDO_UTIL.FROM_WKBGEOMETRY(wkb) -- GML SDO_UTIL.FROM_GMLGEOMETRY(gml2) SDO_UTIL.FROM_GML311GEOMETRY(gml3)	-- Oracle SDO_Geometry SDO_GEOMETRY(SDO_GTYPE, SDO_SRID, SDO_POINT, SDO_ELEM_INFO, SDO_ORDINATES) -- WKT und WKB GET_WKT() SDO_UTIL.TO_WKTGEOMETRY(sdo_geometry) GET_WKB() SDO_UTIL.TO_WKBGEOMETRY(sdo_geometry) -- GML SDO_UTIL.TO_GMLGEOMETRY(sdo_geometry) SDO_UTIL.TO_GML311GEOMETRY(sdo_geometry)
Oracle ST_Geometry (Oracle 2007)	-- Oracle ST_Geometry ST_Geometry(SDO_GEOMETRY) ST_Point(SDO_GEOMETRY) ST_Geometry.FROM_SDO_GEOM(SDO_GEOMETRY) -- WKT und WKB ST_Geometry.FROM_WKT(wkt, [srid]) ST_Geometry.FROM_WKB(wkb, [srid]) OGC_PointFromText(wkt, srid) OGC_PointFromWKB(wkb, srid) -- entsprechende Konstruktoren auch für andere Geometrietypen (z. B. OGC_LineStringFromText(wkt, srid)) -- SQL/MM Spatial Konstruktoren ST_Point(x, y, [srid])	-- Oracle ST_Geometry(SDO_Geometry) ST_Geometry(SDO_GEOMETRY(SDO_GTYPE, SDO_SRID, SDO_POINT, SDO_ELEM_INFO, SDO_ORDINATES)) -- entsprechend auch für Unterklassen, z. B. ST_Point(SDO_GEOMETRY(SDO_GTYPE, SDO_SRID, SDO_POINT, SDO_ELEM_INFO, SDO_ORDINATES)) GET_SDO_GEOM() -- WKT und WKB GET_WKT() GET_WKB() -- Eine Methode zur GML Ausgabe steht noch nicht bereit.

<p>PostgreSQL/PostGIS PG_GEOMETRY (PostGIS 2008b)</p>	<pre>-- entsprechende Konstruktoren auch für andere Geometrietypen (z. B. ST_LINESTRING(koordinaten, [srid])) -- PostGIS EWKT und EWKB ST_GeomFromEWKT(ewkt) (Beispiel EWKT Text: 'SRID=31467;POINT(3500000 5400000)') ST_GeomFromEWKB(ewkb) ST_WKTToSQL(ewkt) ST_WKBToSQL(ewkb) ST_GEOMETRY(ewkt ewkb) -- WKT und WKB GeomFromText(wkt, [srid]) -- OGC SFA 1.1.0 GeomFromWKB(wkb, [srid]) -- OGC SFA 1.1.0 ST_GeomFromText(wkt, [srid]) -- SQL/MM Spatial ST_GeomFromWKB(wkb, [srid]) -- SQL/MM Spatial ST_PointFromText(wkt, [srid]) -- SQL/MM Spatial ST_PointFromWKB(wkb, [srid]) -- SQL/MM Spatial -- entsprechende Konstruktoren auch für andere Geometrietypen (z. B. ST_LineFromText(wkt, [srid])) - Eine Methode zur Konstruktion aus GML steht noch nicht bereit (z. B. ST_GeomFromGML oder ST_GMLToSQL).</pre>	<pre>-- PostGIS EWKT und EWKB ST_AsEWKT(pg_geometry) ST_AsEWKB(pg_geometry,NDR XDR) ST_AsHEXEWKB(pg_geometry,NDR XDR) -- WKT und WKB ST_AsText(pg_geometry) ST_AsBinary(pg_geometry,NDR XDR) -- NDR/XDR: Spezifische Binärcodierung für die WKB Repräsentati- on, entweder in "Little Endian byte order encoding" (NDR) oder in "Big Endian byte order encoding" (XDR) (siehe OGC 2006a). -- GML ST_AsGML([version], pg_geometry, [precision]) (version = GML 2 oder GML 3) -- SVG und KML ST_AsSVG(pg_geometry, [rel], [precision]) ST_AsKML([version], pg_geometry, [precision])</pre>
<p>ESRI ST_GEOMETRY in Oracle (Esri 2008a)</p>	<pre>-- WKT und WKB (SQL/MM Spatial) SDE.ST_GEOMETRY(wkt, srid) SDE.ST_POINT(wkt, srid) oder SDE.ST_POINT(x, y, [m], [z], srid) -- entsprechende Konstruktoren auch für andere Geometrietypen (z. B. SDE.ST_LINESTRING(wkt, srid)) SDE.ST_GeomFromText(wkt, srid) SDE.ST_GeomFromWKB(wkb, srid) SDE.ST_PointFromText(wkt, srid) -- Eine Methode zur Konstruktion aus GML steht noch nicht bereit (z. B. ST_GeomFromGML oder ST_GMLToSQL).</pre>	<pre>-- WKT und WKB (SQL/MM Spatial) SDE.ST_AsText(sde.st_geometry) SDE.ST_AsBinary(sde.st_geometry) -- Eine Methode zur GML Ausgabe steht noch nicht bereit (z. B. ST_AsGML).</pre>

<p>ESRI ST_GEOMETRY in PostgreSQL (Esri 2008a)</p>	<pre>-- WKT und WKB (SQL/MM Spatial) ST_GEOMETRY(wkt, srid) ST_POINT(x, y, [m], [z], srid) entspre- chende Konstruktoren auch für andere Geometriertypen (z. B. ST_LINESTRING(wkt, srid)) ST_GeomFromWKB(wkb, srid) -- Eine Methode zur Konstruktion aus GML steht noch nicht bereit (z. B. ST_GeomFromGML oder ST_GMLToSQL).</pre>	<pre>-- WKT und WKB (SQL/MM Spatial) ST_AsText(st_geometry) ST_AsBinary(st_geometry) -- Eine Methode zur GML Ausgabe steht noch nicht bereit (z. B. ST_AsGML).</pre>
--	--	---