

Geologische Kartierung mit GIS auf der Grundlage von 3D-Modellen*

Martin Breunig¹, Armin B. Cremers¹, Hans-Jürgen Götze²,
Robert Seidemann³, Sabine Schmidt², Serge Shumilov¹ und
Agemar Siehl³

Abstract

We present an integrated approach to support the generation of geological maps by the combined use of 3D geological and geophysical modeling tools. A component based 3D Geo-Information-System is developed, which consists of a geological and a geophysical modeling tool as well as a data handling component. The geoscientific modeling tools have direct access to the database via the Common Object Request Broker Architecture (CORBA), an object-oriented software architecture for distributed software. The benefits of the new approach for geology and geophysics are manifold and related to an eased handling of data and models and the integration of geoscientific methods, all resulting in better validated 3D models and consequently in more accurate and plausible geological maps. The approach is tested within a study area in Southern Lower Saxony.

1 Einleitung

Das Ziel der hier vorgestellten Arbeit ist die Erstellung geowissenschaftlicher Karten durch die Verschneidung geologisch-geophysikalischer 3D-Strukturmodelle mit einem digitalen Höhenmodell (DHM). Die Genauigkeit hinsichtlich der Auflösung der resultierenden Karte richtet sich dabei nach dem zugrundeliegenden Raster des DHM und der Diskretisierung des 3D-Modells. Der Fehler der Karte wird vorrangig von der Rasterweite sowie dem vertikalen Fehler des DHM und der Qualität des 3D-Modells bestimmt. Letzteres wird sowohl durch die Dichte als auch durch die Qualität der zugrunde liegenden geowissenschaftlichen Daten bestimmt.

* Diese Arbeit wird unterstützt von der Deutschen Forschungsgemeinschaft (DFG) innerhalb des DFG-Antragbündels „Interoperable Geowissenschaftliche Informationssysteme“.

¹ Universität Bonn, Institut für Informatik III, email: {martin|abc|shumilov}@cs.uni-bonn.de

² Freie Universität Berlin, Institut für Geologie, Geophysik und Geoinformatik,
email: {hajo|sabine}@geophysik.fu-berlin.de

³ Universität Bonn, Geologisches Institut, email: robert@geo.uni-bonn.de, siehl@uni-bonn.de

Die Konstruktion eines geometrisch-geologischen 3D-Modells aus den zur Verfügung stehenden Daten stellt den Ausgangspunkt dar, wobei die Konstruktion rechnergestützt erfolgt. Hierdurch wird dem Geowissenschaftler eine verbesserte Möglichkeit zur visuellen Überprüfung der vorhandenen Rauminformationen gegeben und es können wichtige Hinweise auf die 'Tauglichkeit' des geowissenschaftlichen Modells geliefert werden.

Die Konstruktion eines geometrisch-geologischen 3D-Modells gestaltet sich allerdings nicht einfach, da die zu erstellenden geologischen Strukturen oftmals äußerst komplex sind und die Datengrundlage ihrerseits sehr häufig und in starkem Maße heterogen ist.

Die Heterogenität der Daten beruht auf mehreren Faktoren: Einerseits auf der räumlichen Verteilung der Daten, da diese aufgrund der verwendeten Quelle nicht immer gleichmäßig in der Fläche oder im Raum verteilt sind, sondern in Gruppen oder Linien angeordnet sind. Andererseits ergibt sich eine Heterogenität aus der Art der Informationsgewinnung, so daß die Daten ohne vorherige Aufbereitung nicht unbedingt kombinierbar sind. Einen dritten Faktor stellt die zeitliche Heterogenität der Daten dar, da man generell davon ausgehen kann, daß die vorliegenden Daten über einen längeren Zeitraum erhoben wurden. Kann man die Schwierigkeiten, die sich durch die beiden zuletzt genannten Punkte ergeben, in den meisten Fällen durch eine Überarbeitung beheben, so bleibt die Problematik, die sich aus einer räumlich ungleichen Verteilung ergibt, erhalten.

Im Rahmen der Konstruktion des 3D-Strukturmodells werden zwei Modellierungsverfahren kombiniert. Die geometrisch-geologische Modellierung erfolgt mit *GOCAD*. Das 3D-Modell ist aus geologischen Grenzflächen (z.B. Schichtgrenzen, Störungsflächen) aufgebaut oder ist ein daraus abgeleitetes 3D-Volumenmodell und muß geologischen Plausibilitätskriterien entsprechen (Siehl 1993). Andererseits wird eine geophysikalische 3D-Modellierung mit *IGMAS* (Götze/Lahmeyer 1988) auf der Basis gravimetrischer Daten durchgeführt. Hierbei wird durch die Parametrisierung der jeweiligen 3D-Volumenmodelle deren Plausibilität nochmals überprüft. Einen wichtigen Aspekt stellt die geometrische und die geowissenschaftliche Konsistenz der 3D-Modelle dar.

Konstruktion und Validierung des 3D-Modells erfolgen durch iterativen Austausch der Ergebnisse der jeweiligen Modellierung. Diese Arbeitsweise erlaubt es, eine höhere geologische Auflösung der 3D-Modelle im Untergrund zu erreichen. Dabei muß diese integrierte Bearbeitung des Untersuchungsgebietes auf heterogenen Plattformen über den bloßen Datentransfer via ftp oder e-mail zwischen den geowissenschaftlichen Applikationen hinausgehen, so daß eine konsistente Datenhaltung der primären und abgeleiteten Daten sowie der Modelle gewährleistet ist. Die Kopplung der geologischen Applikationen (*GOCAD/IGMAS*) an das Geo-Datenbanksystem *GeoToolKit/GeoStore* gewährleistet diese Konsistenz, was besonders hinsichtlich des drastisch zunehmenden Umfangs der zu verwaltenden Datenmenge wichtig ist.

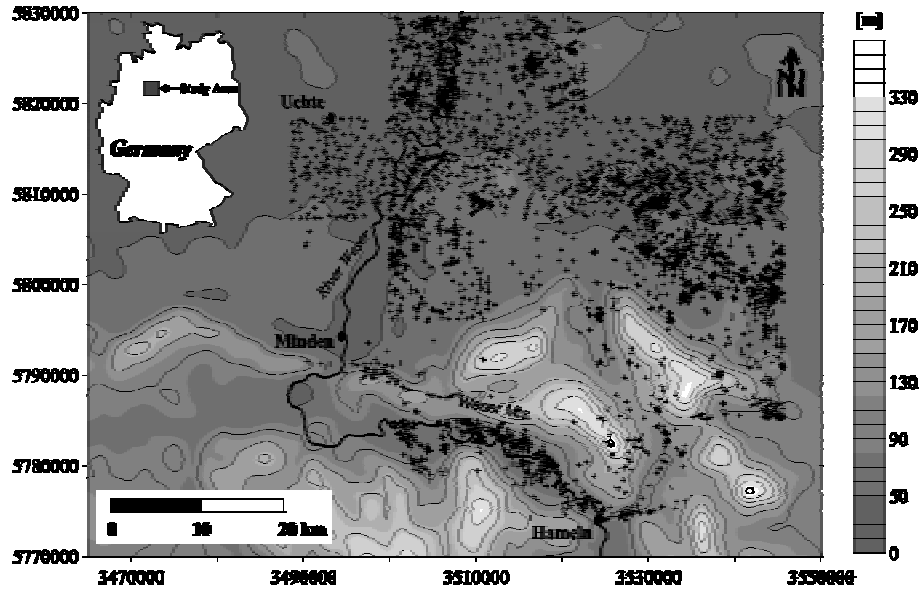


Abbildung 1

Höhenkarte des Untersuchungsgebietes (*Schaumburg-Lippe'sche Kreidemulde*) mit Lage der Bohrungen (+)

Als Untersuchungsgebiet wurde die Schaumburg-Lippe'sche Kreidemulde ausgewählt (Abb.1), die im südlichen Norddeutschen Becken am Nordrand des Wesergebirges liegt. Im Süden und Westen wird das Gebiet von der Weser, im Norden durch das Steinhuder Meer und im Osten entlang einer N-S verlaufenden Linie E' Hannover begrenzt. Wegen der in diesem Gebiet durchgeführten Exploration auf Kohlenwasserstoffe liegt eine hinreichend große Datendichte vor, welche eine wichtige Voraussetzung für die Durchführung der Modellierung ist. Es handelt sich hierbei um über 5000 Bohrungen sowie 14 Isolinienpläne und 4 Profilschnitte (Kockel et al. 1996). Darüberhinaus stehen Datensätze gemessener Schwere- und Magnetfelder und Informationen seismischer Untersuchungen und Bohrlochgeophysik zur Verfügung (Breunig et al. 1999).

Das Wesergebirge als südliche Begrenzung des Untersuchungsgebietes bietet mit seinem ausgeprägten Relief gute Aufschlußverhältnisse, so daß es möglich ist, die rechnergestützt erzeugte geologische Karte im Gelände zu validieren.

2 Verwendete Applikationen

2.1 Geowissenschaftliche Werkzeuge (GOCAD und IGMAS)

GOCAD ist ein Programm zur interaktiven Konstruktion und Visualisierung geologischer Modelle aus Punkt-, Linien- und/oder Flächendaten (Mallet 1992a, GOCAD 1999). Die Verwendung des robusten Interpolations- und Approximationsalgorithmus *DSI* (*discrete smooth interpolation*) (Mallet 1992b) ermöglicht unter Berücksichtigung interaktiv zu spezifizierender Randbedingungen eine flexible Generierung und Modifizierung der 3D-Geometrieobjekte.

IGMAS ist ein Programm, mit dem gemessene Potentialfelder der Gravimetrie und/oder der Magnetik durch numerische 3D-Modellrechnungen interpretiert werden können. Hierbei wird die Struktur des Untergrundes in Volumina untergliedert und diesen Volumina ein Dichte- und/oder Suszeptibilitätswert zugewiesen. Das Verfahren ist ein interaktives Optimierungsverfahren, bei dem der Bearbeiter einerseits die Geometrie, andererseits die Parametrisierung der Gesteinseigenschaften ändert. Hierfür steht ein GUI mit umfangreicher Funktionalität zur Verfügung. Die Methode basiert auf der Anwendung des Gauß'schen Satzes sowie der Green'schen Formeln zur Lösung des Volumen-Integrals, das auf ein Oberflächenintegral zurückgeführt werden konnte (Götze 1978, Götze 1984, Götze/Lahmeyer 1988).

2.2 Datenbankwerkzeug (GeoToolKit)

Das Datenbankwerkzeug *GeoToolKit* (Balovnev et al. 1997a, GeoToolKit 1999) ermöglicht eine effiziente Speicherung und Abfrage räumlicher 3D-Objekte in einer Datenbank. Hierzu ist *GeoToolKit* über *GeoStore* eng mit dem objektorientierten DBMS *ObjectStore* gekoppelt. Abb.2 zeigt das Datenmodell von *GeoToolKit* für persistente Objekte in einer OMT-ähnlichen Notation (Rumbaugh et al. 1991). Eine abstrakte *SpatialObject* Klasse – die Wurzel der räumlichen Objektklassenhierarchie – spezifiziert eine Schnittstelle, die von den konkreten räumlichen Objekten vererbt wird. Eine konkrete Klasse stellt eine Repräsentation sowohl für das Objekt als auch eine Implementierung für geometrische Funktionen bereit. Daher garantiert *GeoToolKit*, daß jedes räumliche Objekt mindestens die Funktionalität der allgemeinsten Klasse besitzt. Gewöhnlich haben konkrete Klassen weitere Funktionen, die dieser geometrischen Entität eigen sind.

Augenblicklich stellt *GeoToolKit* verschiedene Klassen für die Repräsentation und Manipulation von einfachen (Punkt, Segment, Dreieck, Tetraeder) und komplexen (Linie, Fläche, Volumen) räumlichen 3D-Objekten zur Verfügung. Da die *GeoToolKit*-Klassenhierarchie vollständig ist, kann jedes räumliche Objekt entweder direkt, durch eine der räumlichen *built-in*-Klassen oder als Komposition dieser Klasse in einer Gruppe (*Group*) modelliert werden. Eine Gruppe ist eine heterogene Kollektion räumlicher Objekte, die weiterhin als einzelne Objekte behandelt werden.

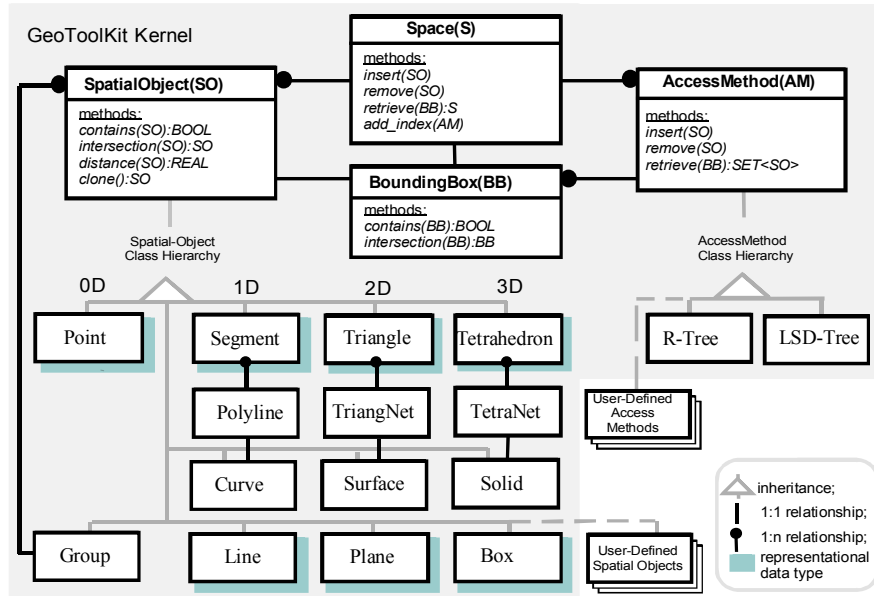


Abbildung 2
Das Objektmodell von *GeoToolKit*

Jeder neu definierte Datentyp, der gemäß den *GeoToolKit*-Konventionen genutzt wird, kann in die Klassenbibliothek zur weiteren Verwendung eingebunden und auch wiederverwendet werden. Ein neuer Typ kann die Repräsentation und die Funktionalität der eingebetteten Klassen erben und dabei Funktionen redefinieren.

Reale geologische Strukturen sind in den geowissenschaftlichen Modellierungsprogrammen oftmals nur durch extrem komplexe, nicht-reguläre Formen abzubilden, die in eine endliche Menge räumlicher Primitive zerlegt werden können. Entsprechend bietet *GeoToolKit* die Klassen *Polyline*, *TriangleNet* und *TetraNet* als Standard-Repräsentationen für die abstrakten Klassen *Curve*, *Surface* bzw. *Solid*.

GeoToolKit unterscheidet zwischen geometrischen Prädikaten, die *true* oder *false* (*equal*, *intersect*, *contains*) zurückliefern, und geometrischen Operationen (*intersection*, *division*), die ein räumliches Objekt zurückliefern, das in der Datenbank abgespeichert oder als Argument in anderen geometrischen Operationen genutzt werden kann.

Space stellt eine besondere Klasse (*container class*) dar, die für eine effiziente Suche der Elemente nach ihrer Lage im Raum zuständig ist und entweder exakt durch einen Punkt oder eine Begrenzung (*bounding box*) spezifiziert ist. Darüber hinaus dient diese Klasse auch als Programmschnittstelle für den Manager der räumlichen Abfragen.

3 Kopplung von GOCAD und IGMAS mit GeoToolKit/GeoStore

Die Kopplung der geowissenschaftlichen 3D-Modellierungstools *GOCAD* (Mallet 1992a, GOCAD 1999) und *IGMAS* (Götze/Lahmeyer 1988) mit der Geo-Datenbank *GeoToolKit/GeoStore* erfolgt auf der Basis objektorientierter Softwaretechnologien (CORBA).

Der implementierte Prototyp verwendet CORBA um einen transparenten Zugriff in einem heterogenen Netzwerk mit unterschiedlichen Plattformen (Sun Solaris, HP UX, SGI Irix und WindowsNT) zu ermöglichen. *GeoToolKit/GeoStore* ist von einem Wrapper umgeben, so daß *GeoToolKit/GeoStore* als eine verteilte Geo-Datenbank angesehen werden kann (Balovnew et al. 1997b, Balovnew et al. 1998). Dabei ermöglicht der Wrapper einen einfachen Zugriff der CORBA Klienten (*GOCAD* und *IGMAS*) auf die Datenbank, setzt jedoch deren Kompatibilität zu CORBA voraus (Abb. 3).

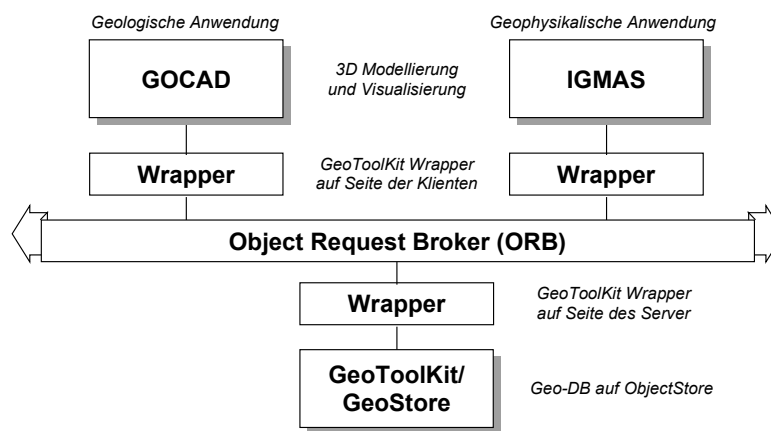


Abbildung 3
Generelles Schema der 3D-GIS Architektur

Die Vorteile einer Kopplung von *GOCAD* und *IGMAS* mit *GeoToolKit/GeoStore* sind offensichtlich:

1. Mit den geowissenschaftlichen Applikationen (*GOCAD* bzw. *IGMAS*) erzeugte oder bearbeitete geowissenschaftliche Daten und Modelle können direkt in der OODB gespeichert sowie auch angefordert werden.
2. Daten und Modelle, die aus einer Applikation (z.B. *GOCAD*) heraus in der OODB eingelagert wurden, können mit einer anderen Applikation (z.B. *IGMAS*) wieder ausgelesen werden.

3. Die in der OODB implementierten Methoden können auch von den geowissenschaftlichen Applikationen genutzt werden.
4. Die Ergebnisse der Datenbankabfragen können mit den geowissenschaftlichen Applikationen visualisiert und weiter bearbeitet werden.
5. Die Konsistenz der Daten und Modelle im Hinblick auf Versionen und Zugriffe ist gewährleistet.

3.1 Integration des OODBMS mit CORBA

In einem ersten Ansatz wurde die Integration eines OODBMS mit ORB am Beispiel der ORB-Implementation von Orbix (IONA Technologies 1997a) und des OODBMS ObjectStore (Object Design Inc. 1997) evaluiert.

Die CORBA 2.0/IIOP Spezifikation (Object Management Group 1998a) definiert zwei Standards zur Integration: zum einen den *Persistent Object Service (POS)* und zum anderen den *Object Database Adapter (ODA)*.

Die Definition von *POS* ist sehr allgemein, gleichzeitig jedoch komplex in der Implementierung (Object Management Group 1998b). Einige Lücken in der Spezifikation erschweren eine Implementation (Kleindienst et al. 1996), weshalb es zum jetzigen Zeitpunkt keine vollständige Implementierung gibt. Aus diesem Grund wird dieser Ansatz im Folgenden nicht weiter betrachtet.

Die Spezifikation des *ODA* ist dagegen vollständig und definiert einen Objekt-Adapter für Objekte, die in einer Datenbank gespeichert sind. Der *ODA* verwaltet den persistenten Status des Implementations-Objektes und ist darauf ausgerichtet deren Programmierung zu vereinfachen. Die Implementations-Objekte werden in der Datenbank gespeichert und dynamisch in den Speicher des Servers geladen sobald eine Anfrage gestartet wird. Im Gegensatz zum *Basic Object Adapter (BOA)* vorhergehender CORBA Spezifikationen nimmt der *ODA* nicht an, daß sich jede Implementation im Speicher befindet. Darüberhinaus sollte der *ODA* so implementiert sein, daß weitere typische Merkmale einer Datenbank, wie beispielsweise Sperren und Transaktionen, möglich sind, da auf persistente Objekte generell von allen Seiten zugegriffen werden kann.

Abgesehen von diesen Standardansätzen gibt es eine große Zahl weiterer, nicht standardisierter Ansätze, wobei bei den meisten die Nutzung spezieller Software nicht vorausgesetzt wird, jedoch den Eingriff eines Programmierers erfordern. Üblich ist hier die Verwendung einer Verbindungsschicht (*middle-tier-layer*), die die originale Schnittstelle der Datenbank umhüllt und eine äquivalente, CORBA compatible Schnittstelle zur Verfügung stellt. Diese Schicht wird als *Wrapper* bezeichnet und auf diesem Prinzip beruhende Ansätze als *Wrapper-Ansätze*.

Um Datenmodelle und Methoden der OODB den Klienten als CORBA Objekte zugänglich zu machen, ist es notwendig, daß ein Wrapper jede Klasse der Datenbank durch eine äquivalente, CORBA-kompatible Klasse abbildet, so daß diese

Klasse jeweils alle entsprechenden Methoden besitzt. Die Instanzen dieser Klassen verhalten sich dabei sowohl wie gewöhnliche CORBA Objekte, als auch wie Klienten der Datenbank. Indem sie die Parameter der Daten konvertieren und alle Funktionsaufrufe in beide Richtungen übertragen vermitteln sie somit zwischen den CORBA-*Skeletons* und den Objekten der Datenbank (Wiederhold 1997). Aufgrund dieser Funktion werden solche CORBA Objekte als *Mediatoren* bezeichnet und entsprechend ein solcher Ansatz als *Mediator-Ansatz*.

Dieser Ansatz wurde weiter verfolgt und mündete in der Entwicklung eines ODA-Prototypen sowie eines speziellen räumlichen ODA (*SODA [spatial ODA]*). Eine detaillierte Beschreibung gibt Shumilov (1999).

3.2 Vorgeschlagener Ansatz

In den Fällen, in denen die Integration mit einer schon existierenden Datenbank notwendig ist, ist ein ODA-Ansatz nicht flexibler als ein Wrapper-Ansatz. Aus diesem Grund verwenden wir einen Ansatz bei dem ein ODA die Grundlage bildet und mit einem Mediatoren-Ansatz verknüpft wird.

Da die Entwicklung und der Aufbau einer Datenbank sehr zeitaufwendig ist, sollte die CORBA-Anbindung idealerweise ohne eine Änderung der Struktur der Datenbank erreichbar sein. Aus diesem Grund sollten die Objekte der Datenbank für die CORBA-Klienten mittels korrespondierender, transienter Mediatoren zugreifbar sein. Die persistenten Objekte der Datenbank, und damit auch die Daten, werden lediglich über Schnittstellen freigegeben. Die Mediatoren kapseln sowohl die Daten, als auch die Methoden zur Manipulation der Daten. Der Klient kann dynamisch über die Mediatoren auf die OODB zugreifen, was durch die Anwendung anderer Methoden nur schwer zu erreichen ist.

Die Transienz der Mediatoren hat mehrere Vorteile. Jedes CORBA-Objekt besteht aus einem Datenbank-Objekt und einem korrespondierendem Mediator. Solange die Mediatoren transient sind, verändern sie nicht die originale Datenbank und ermöglichen dem ODA alle Arten von Transaktionen, die von dem OODBMS angeboten werden, ebenfalls zu unterstützen. Allerdings muß die Lebensdauer der Mediatoren kontrolliert werden. Zudem muß gewährleistet sein, daß mehrere, gleichzeitig existierende Mediatoren nicht mit demselben Datenbank-Objekt verknüpft sind.

In der hier vorgeschlagenen ODA-Architektur werden die Mediatoren von speziellen Managern in Form verknüpfbarer ODA-Module kontrolliert. Der Kernel des Adapters besteht aus fünf Objekten (Abb.4): Einem *Wrapper-Objekt*, welches die Schnittstelle der vier weiteren, austauschbaren Komponenten darstellt; einem Lader (*loader*), der die persistenten Objekte erhält und, wenn dies notwendig wird, neue Mediatoren erzeugt; einem Datenbank Manager (*database manager*), der eine Instanz der *ODA_DatabaseManager*-Klasse ist und alle Mediatoren des/der OODBMS

kontrolliert, die wiederum Instanzen der `ODA_Database`-Klasse darstellen; einem Typ Manager (*type manager*), der eine Instanz der `ODA_TypeManager`-Klasse ist und alle zur Verfügung stehenden Mediator-Klassen verwaltet und einem Transaktions Manager (*transaction manager*), der eine Instanz der `ODA_TransactionManager`-Klasse ist und Anfang und Ende der OODB-Transaktionen kontrolliert.

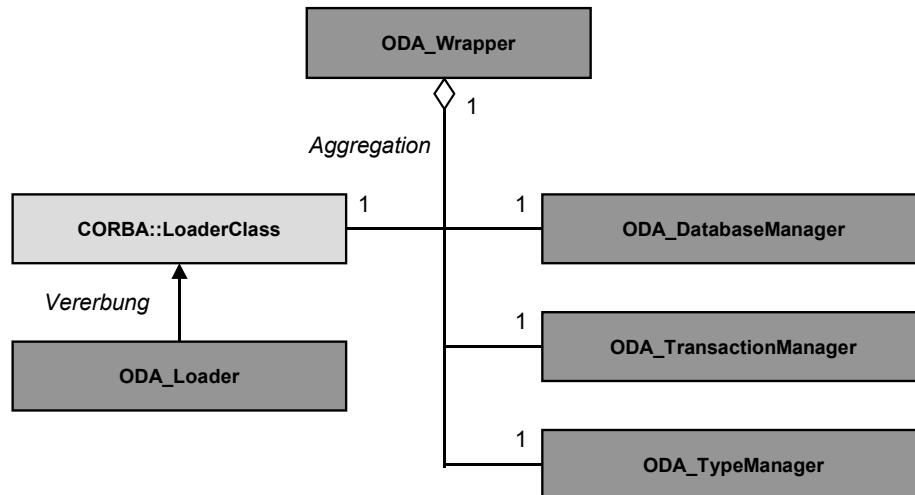


Abbildung 4
Struktur der Klassen des Object Database Adapters (ODA)

3.3 Funktionalität des ODA

Die wichtigsten Aspekte hinsichtlich der Funktionalität des ODA sind:

1. Die Methode mit der die Objekte einer OODB angebunden und somit innerhalb der CORBA-Umgebung verfügbar sind.
2. Die Repräsentation der Objekt Referenzen der OODB innerhalb der CORBA-Umgebung.
3. Die Aktivierung und Deaktivierung der Objekte innerhalb der CORBA-Umgebung.
4. Die Transaktionskontrolle, durch die alle persistenten Objekte lediglich im Rahmen einer Transaktion erreichbar sind.

5. Die Kontrolle der OODB innerhalb der CORBA-Umgebung, bei der der ODA eine Schnittstelle zur Verfügung stellt, welche die gesamte Funktionalität der OODB zur Verfügung stellt.

Die vorgestellte, auf einem Mediator-Ansatz beruhende ODA-Architektur stellt einen effektiven Weg dar ein OODBMS innerhalb eines verteilten Systems bereitzustellen, wobei die Struktur existierender OODBs nicht geändert werden muß. Die Anpassungsfähigkeit dieses ODA wurde am Beispiel eines SODA getestet, der eine spezielle Implementation für *GeoToolKit/GeoStore* darstellt. Dabei konnte durch die Wiederverwendung des ODA innerhalb des SODA der Umfang des Quellcodes auf der Seite des Server eingeschränkt werden.

4 Ergebnisse

Durch die Möglichkeit einer kontrollierten, iterativen Modellvalidierung via ORB ergeben sich für die Geowissenschaften Vorteile, die sich in vier Bereiche gliedern lassen.

Ein generelle Verbesserung ergibt sich aus der Nutzung einer OODB (*GeoToolKit/GeoStore*) im Verlauf der Datenaquisition und 3D-Modellierung. Die zentrale Datenhaltung ermöglicht es, daß alle verfügbaren Daten den Bearbeitern aller beteiligten Arbeitsgruppen zur Verfügung stehen. Gleichzeitig wird auch die Aktualisierung des Datenbestandes, wie die Einbindung neuer Daten oder auch die Überarbeitung vorhandener Daten vereinfacht. Ebenso wird die Haltung und der Zugriff auf die erstellten Modellversionen erleichtert. Dies beinhaltet auch die Nutzung spezieller DB-Funktionen wie Zugriffs- und Bearbeitungsprotokolle, Warnungen und Verbote der zeitgleichen Nutzung derselben Daten, durch die gewährleistet wird, daß keine parallele Modellentwicklung stattfindet.

Durch die direkte Kopplung der geowissenschaftlichen Applikationen an die OODB mittels eines ORB werden die schon angesprochenen Vorteile, die sich aus der Nutzung der OODB ergeben, nochmals verbessert. Darüber hinaus führt die damit einhergehende Integration verschiedener geowissenschaftlicher Methoden zum Aufbau eines interoperablen 3D-GIS.

Der direkte und gemeinsame Zugriff auf die in der OODB eingelagerten Daten und Modelle erleichtert den Austausch der 3D-Modelle zwischen den verschiedenen Applikationen deutlich. Dies gilt besonders für räumlich verteilte Arbeitsgruppen, da davon auszugehen ist, daß diese in einem verteilten und heterogenen Netzwerk arbeiten. Darüber hinaus ist die zukünftige Erweiterbarkeit des interoperablen GIS um weitere räumlich verteilte Datenbanken und Applikationen gewährleistet. Da die Kopplung plattformunabhängig erfolgt, kann die eingebundene Software nahezu unverändert weiter genutzt werden.

Ein weiterer Vorteil, der sich durch die Funktionalität des 3D-GIS ergibt, stellt die gemeinsame Nutzung eines Archivs von Standardmethoden dar, so daß die verschiedenen Modellierungstools auf gemeinsame Methoden zurückgreifen können.

Letztlich führt die Nutzung eines interoperablen 3D-GIS durch die Integration der geowissenschaftlichen Applikationen zu einer deutlichen Verbesserung der Interpretationsmöglichkeit komplexer geowissenschaftlicher Sachverhalte. Aus der Kopplung der geowissenschaftlichen Applikationen und dem dadurch erleichterten iterativen Austausch der jeweiligen Ergebnisse der Modellierung ergibt sich die Möglichkeit zur rigorosen Validierung des Gesamtmodells, da mehr Stützstellen zur Verfügung stehen. Daraus folgt letztlich eine verbesserte Plausibilität der geologischen Karte, welche aus dem Schnitt des 3D-Modells mit dem DHM resultiert.

Literaturverzeichnis

- Alms, A., Balovnev, O., Breunig, M., Cremers, A.B., Jentzsch, T., Siehl, A. (1998): Space-Time Modelling of the Lower Rhine Basin Supported by an Object-Oriented Database, in: *Physics and Chemistry of the Earth*, 23, No. 3, S. 251-260
- Balovnev, O., Breunig, M., Cremers, A.B. (1997a): From GeoStore to GeoToolKit: The Second Step, in: *Proceedings of the 5th Intern. Symposium on Spatial Databases*, Berlin, LNCS No. 1262, Berlin, S. 223-237
- Balovnev, O., Breunig, M., Cremers, A.B., Shumilov, S. (1997b): GeoToolKit: Opening the Access to Object-Oriented Geodata Stores, in: *International Conference on Interoperating Geographic Information Systems (InterOp'97)*, Santa Barbara, CA, USA, www.informatik.uni-bonn.de/~shumilov/publications/papers/interop97/abstract.html
- Balovnev, O., Breunig, M., Cremers, A.B., Pant, M. (1998): Building Geo-Scientific Applications on Top of GeoToolKit: A Case Study of Data Integration, in: *Proceedings of the 10th Intern. Conference on Scientific and Statistical Database Management*, IEEE Computer Society, Los Alamitos/CA, Capri, 1-3 July 1998, S. 260-269
- BEA Systems, International Business Machines Corporation, Iona Technologies, Netscape Communications Corporation, Novell, Visigenic Software (1998): *Objects by Value Joint Revised Submission*. orbos/98-01-18
- Breunig, M., Cremers, A.B., Götze, H.-J., Schmidt, S., Seidemann, R., Shumilov, S., Siehl, A. (1999): First Steps Towards an Interoperable GIS - An Example From Southern Lower Saxony, in: *Physics and Chemistry of the Earth*, 24, No. 3, S. 179-190
- GeoToolKit (1999): *Technical Documentation*, www.cs.uni-bonn.de/III/projekte/d4/gtk/index.htm
- Götze, H.-J. (1978): Ein numerisches Verfahren zur Berechnung der gravimetrischen Feldgrößen drei-dimensionaler Modellkörper, in: *Arch. Met. Geoph. Biokl., Ser. A*, 25, Wien, S. 195-215
- Götze, H.-J. (1984): *Über den Einsatz interaktiver Computergraphik im Rahmen 3-dimensionaler Interpretationstechniken in Gravimetrie und Magnetik*, Habil. Schrift, TU Clausthal

- Götze, H.-J., Lahmeyer, B. (1988): Application of Three-Dimensional Interactive Modeling in Gravity and Magnetism, in: *Geophysics*, 53, No. 8, S. 1096-1108
- GOCAD (1999): Technical Documentation, www.ensg.u-nancy.fr/GOCAD
- IONA Technologies Ltd. (1997a): *Orbix Programmer's Guide*, Version 2.3
- IONA Technologies Ltd. (1997b): *Orbix+ObjectStore Adapter Programming Guide*
- Kleindienst, J., Plasil, F., Tuma, P. (1996): What We Are Missing in the Persistent Object Service Specification, *OOPSLA'96 Workshop on Large Persistent and Distributed Systems*, <http://nenya.ms.mff.cuni.cz/thegroup/rep/work0908.ps.Z>
- Kockel, F. et al. (1996): *Geotektonischer Atlas von Nordwest-Deutschland*, BGR, 4S., 16 Karten, 8 Profilblätter, Hannover
- Mallet, J.L. (1992a): GOCAD: A Computer Aided Design Program for Geological Applications, in: Turner, A.K. (Hrsg.): *Three-Dimensional Modeling with Geoscientific Information Systems*, NATO ASI 354, Dordrecht, S. 123-142
- Mallet, J.L. (1992b): Discrete Smooth Interpolation in Geometric Modelling, in: *Computer Aided Design*, 24, No. 4, S. 178-192
- Object Design Inc. (1997): *ObjectStore C++ API User Guide*, Release 5.0
- Object Management Group (1998a): CORBA 2.0/IIOP Specification, OMG formal document 98-07-01. <http://www.omg.org/corba/c2indx.htm>
- Object Management Group (1998b): CORBA Services: Common Object Services Specification, OMG formal document 98-12-09, <http://www.omg.org/corba/csindx.htm>
- Rumbaugh et al. (1991): *Object-Oriented Modeling and Design*, New Jersey
- Schmidt, S., Götze, H.-J. (1999): Integration of Data Constraints and Potential Field Modelling - An Example From Southern Lower Saxony, in: *Physics and Chemistry of the Earth*, 24, No. 3, S. 191-196
- Shumilov, S. (1999): Mediator-Based Object Database Adapter Architecture, Proposed to: International Symposium on Distributed Objects and Applications, Edinburgh, Scotland.
<http://www.informatik.uni-bonn.de/~shumilov/publications/papers/doa99/doa99.ps.gz>
- Siehl, A. (1993): Interaktive geometrische Modellierung geologischer Flächen und Körper, in: *Die Geowissenschaften*, Heft 10-11, S. 342-346
- Swainston-Rainford, M. (1997): CORBA Objects by Value
<http://www.mjs-r.com/RESOURCE/idldesign/objectsbyvalue.html>
- Wiederhold, G. (1997): Value-Added Mediation in Large-Scale Information Systems, in: Meersman, R., Mark, L. (Hrsg.): *Database Application Semantics*, London et al., S. 34-56, <http://www-db.stanford.edu/pub/gio/1995/DS6CH.ps>