

# **GODOT: Ein objektorientiertes Geoinformationssystem**

Jürgen Ebbinghaus, Günter Hess, Joachim Lambacher,  
Wolf-Fritz Riekert, Thomas Trotzki, Gerlinde Wiest<sup>1</sup>

## **1. Einleitung**

In dem FAW-Forschungsprojekt GODOT (Geodatenhaltung mit objektorientierten Techniken) wird der Prototyp eines objektorientierten Geoinformationssystems (GIS) konstruiert, das direkt auf einem kommerziellen objektorientierten Datenbanksystem (OODBS) aufsetzt (Riekert 1993). Auftraggeber für GODOT sind das Umweltministerium und das Innenministerium Baden-Württemberg sowie die Siemens Nixdorf Informationssysteme AG und die Siemens AG. Als Anwendungsgebiet des entwickelten GIS steht die Abbildung komplexer Umweltsachverhalte im Vordergrund, wie sie in den verschiedenen Komponenten des Umwelteinformationssystems Baden-Württemberg (UIS) verwaltet werden. Die Komplexität bezieht sich dabei nicht nur auf die Geometrie- und Topologiedaten, sondern schließt auch komplex strukturierte Sachdaten und vielfältige semantische Beziehungen zwischen den Objekten in Geo- und Umwelthanwendungen ein.

Moderne am Markt erhältliche Geoinformationssysteme nutzen relationale Datenbanksysteme zur integrierten Ablage dieser Geometrie- und Sachdaten. Darüber hinaus ist die Nutzung neuer, sogenannter postrelationaler Datenbanksysteme (ACM 1991, Küspert 1994) Gegenstand der Forschung. Insbesondere die seit einigen Jahren auf dem Markt verfügbaren objektorientierten Datenbanksysteme versprechen zusätzliche Vorteile durch weitreichendere Konzepte für die Repräsentation von raumbezogener Information in themenübergreifenden Geo- und Umwelthanwendungen (Ebbinghaus 1994).

Der in GODOT gewählte Ansatz der Erweiterung eines OODBS um geospezifische Datentypen (Klassen) und Operatoren (Methoden) ermöglichte es, den Entwurf und die Implementierung des GODOT-Datenmodells in einer durchgängig objektorientierten Weise durchzuführen. Das resultierende objektorientierte Datenmodell des GODOT-Prototypen kann durch benutzerdefinierte Klassen und Methoden erweitert werden, um verschiedenartigste Anwendungen zu unterstützen. Der GODOT-Prototyp hat als

---

<sup>1</sup>Forschungsinstitut für anwendungsorientierte Wissensverarbeitung an der Universität Ulm, Helmholtzstr. 16, D-89081 Ulm, Email: ebbingha@faw.uni-ulm.de

Basis das kommerzielle OODBS ObjectStore (Lamb u.a. 1991), das als Nicht-Standard-Datenbanksystem die vollständig integrierte Verwaltung sowohl von räumlichen Daten, als auch von komplex strukturierten Sachdaten ermöglicht. Durch die Verwendung eines bereits am Markt etablierten OODBS kann der Fortschritt dieser Systeme u.a. im Bereich Transaktionsmanagement, graphisch-interaktiver Tools und verteilter Verarbeitung sowie die aufkommende Standardisierung (z.B. der Abfragesprache) ausgenutzt werden und die Notwendigkeit von Eigenentwicklungen minimiert werden.

## **2. Objektorientierte Konzepte für Geoinformationssysteme**

In OODBS wird ein Objekt, durch einen vom Datenbanksystem selbständig vergebenen Objektidentifikator über seine gesamte Lebenszeit eindeutig gekennzeichnet. Objektidentifikatoren entlasten die Anwendungen von der Vergabe und der Konsistenzsicherung von Schlüsselattributwerten, insbesondere dort, wo für Geo- und Umweltdaten keine allgemeingültigen Schlüsselssysteme existieren. Die Vergabe der Objektidentifikatoren durch das OODBS stellt sicher, daß auch in einer verteilten Umgebung, in der verschiedene Anwendungen auf mehrere gemeinsame Datenbanken zugreifen, jedes Objekt unabhängig vom Ort seiner Erzeugung systemweit eindeutig identifizierbar ist. Mit Hilfe von Objektidentifikatoren kann auf gemeinsam genutzte Unterobjekte, z.B. auf einen Grenzlinienzug zwischen zwei Flächen-Objekten, der als Unterobjekt zum Umriß beider Flächen gehört, in natürlicher Weise Bezug genommen werden. Änderungen in der Geometrie des Linienzugs wirken sich unmittelbar auf die Umrisse der zugehörigen Flächen-Objekte aus, die durch den gemeinsamen Linienzug als Nachbarflächen identifiziert werden können.

Die in Geo- und Umwelthanwendungen häufig zu behandelnden komplexen Objektstrukturen können in OODBS durch beliebig ineinander verschachtelbare Tupel-, Listen-, und Mengen-Datentypen aus elementaren Datentypen (z.B. Integer, Character) direkt nachgebildet werden, was die Modellierung von Anwendungen stark vereinfacht. Grundsätzlich lassen sich zwei verschiedene Beziehungstypen für den Aufbau komplexer Objekte unterscheiden. Ein Objekt kann als Unterobjekt ein integraler Bestandteil eines anderen Objekts sein (embedded object), wie z.B. die Flußabschnitte Bestandteil eines Flusses sind. Dies bedeutet, daß das Unterobjekt nicht unabhängig von seinem umschließenden Objekt existieren und auch nicht von mehreren Objekten als gemeinsames Unterobjekt verwendet werden kann. Andererseits gibt es Objekte, die für sich existieren und gleichzeitig zu anderen Objekten in Beziehung stehen können (linked object). Verweise zwischen eigenständigen Objekten können bidirektional sein, das heißt, die Objekte "kennen sich gegenseitig". So kann z.B. für einen Landkreis die

Menge der Verweise auf die zugehörigen Gemeinden verwaltet werden und gleichzeitig für jede Gemeinde ein Verweis auf den Landkreis als übergeordnete Verwaltungseinheit existieren. Diese bidirektionalen Verweise werden vom OODBS konsistent gehalten, indem bei Änderungen des Verweises in der einen Richtung auch der Rückverweis aktualisiert wird (z.B. wenn eine Gemeinde durch eine Gebietsreform einem anderen Landkreis zugeordnet wird).

Durch benutzerdefinierte Datentypen und Methoden kann die Funktionalität eines GIS über die vom OODBS vorgegebenen Datentypen (built-in data types) hinaus erweitert und den anwendungsspezifischen Bedürfnissen angepaßt werden. Es ist möglich, die internen Repräsentationen und Zugriffsmethoden so zu wählen, daß für eine Anwendung typische Abfragen und Analysen auf effiziente Weise durchgeführt werden können. So sind z.B. im GODOT-Prototypen die geometrischen Datentypen, die die Lage und räumliche Ausdehnung von geographischen Objekten repräsentieren, mit einer Menge von Integer-Werten versehen, die nach dem Z-Ordering Schema (Orenstein u.a. 1988) ermittelt werden. Diese sogenannten Z-Values stellen eine geometrische Vergrößerung dar und erlauben es bei räumlichen Abfragen, eine indexunterstützte Vorselektion durchzuführen.

In GIS-Anwendungen können Klassenhierarchien mit Vererbung sehr effizient genutzt werden, um die verschiedenen Arten geographischer Objekte mit ihren zugehörigen Sachattributen und Auswertungsfunktionen so zu strukturieren, daß Redundanzen bei den Attribut- und Methodendefinitionen weitgehend vermieden werden. Im GODOT-Prototypen sind z.B. die Datenstrukturen und Funktionen, die für die Verwaltung und Analyse der Lage und räumlichen Ausdehnung von geographischen Objekten (kurz auch Geoobjekte genannt) notwendig sind, in einer allgemeinen Oberklasse (GodotGeoObject) definiert und werden an alle Unterklassen geographischer Objekte vererbt (siehe Abb. 1). Änderungen und Erweiterungen der Funktionalität für räumliche Analysen werden damit auf eine Klasse beschränkt.

Da Geo- und Umweltinformationssysteme eine Vielzahl von verschiedenen Anwendungen unterstützen sollen, bekommt der Aspekt der Erweiterbarkeit und Wartbarkeit von Anwendungen zunehmend Bedeutung. Erweiterungen und Änderungen sollen auch vom Benutzer (bzw. Anwendungsentwickler) durchgeführt werden können, um eine GIS-Anwendung den sich ändernden Anforderungen anzupassen. Die Konzepte der Einkapselung und Polymorphie bieten hierfür wesentliche Hilfestellungen. Am Beispiel der Geometrieklassen des GODOT-Prototypen soll dies im folgenden veranschaulicht werden. Zur Verwaltung der Lage und räumlichen Ausdehnung von geographischen Objekten werden im GODOT-Prototypen vielfältige Klassen von sogenannten Geometrieobjekten verwendet (siehe Abb. 2). Der Entwickler einer Geo- oder Umweltanwendung kann diese Klassen verwenden, um die geometrischen

Eigenschaften der Objekte seiner Anwendung zu repräsentieren, ohne daß er über die Details der Implementierung dieser Klassen genaue Kenntnis haben muß. So bleibt ihm z.B. verborgen, daß die Klassen über eine Menge von Z-Values (siehe oben) verfügen, um räumliche Abfragen effizient durchführen zu können. Was er verwendet, sind die Operationen (Methoden), die ihm die Schnittstelle dieser Klassen zur Verfügung stellt, um z.B. die Distanz zwischen zwei Geobjekten oder das räumliche Enthaltensein von Geobjekten abzufragen. Diese Methoden sind in den einzelnen Geometrieobjekt-Klassen jeweils unterschiedlich implementiert, ihre Namen (z.B. distance, contains) sind aber über alle Klassen hinweg einheitlich gewählt. Der Aufruf einer räumlichen Operation (z.B. distance) hat daher stets dieselbe Syntax, unabhängig von den Klassen der beteiligten Geometrieobjekte. Dies ist von entscheidender Bedeutung für die Erweiterbarkeit des Systems. So müssen bei einer Erweiterung der Geometrieklassen lediglich die entsprechenden Schnittstellen-Methoden (z.B. die Distanzberechnung) für die hinzukommende Geometrieklasse (z.B. Spline) geschrieben werden, damit die Anwendungen die hinzugefügte Geometrieklasse in gleicher Weise verwenden können wie die bereits existierenden. Anpassungen in den Anwendungen sind in der Regel nicht notwendig, da die räumlichen und topologischen Operationen (Methoden) auch für die neue Klasse zur Verfügung stehen und bei Auswertungen (z.B. Distanzberechnung zwischen den entsprechenden Geobjekten) automatisch ausgewählt werden.

### **3. Daten- und Funktionsstrukturierung in GODOT**

Der GODOT-Prototyp wird durch die drei Objektbereiche der *Sachobjekte*, der *Geometrieobjekte* und der *Darstellungsobjekte* strukturiert, die jeweils spezifische Objektklassen mit zugehörigen Funktionen umfassen. Zwischen diesen drei Teilbereichen existieren klar definierte Schnittstellen, die es erlauben, Änderungen und Erweiterungen in einem der Teilmodelle vorzunehmen, ohne daß dadurch Anpassungen in den anderen Bereichen notwendig werden. Der modulare Aufbau des GODOT-Prototyps ermöglicht darüber hinaus die unabhängige Nutzung der Funktionalität von Teilbereichen. So sind z.B. für die Anfragekomponente nur die Sachobjekte bzw. Geobjekte mit ihren zugehörigen Sachattributen und Geometrien von Belang. Andererseits können die erzeugten Darstellungsobjekte als Karten in eigenen Kartographie-Datenbanken gespeichert und mit einer Graphikkomponente visualisiert und modifiziert werden. Im folgenden soll auf die Klassen und Funktionen der drei Objektbereiche näher eingegangen werden.

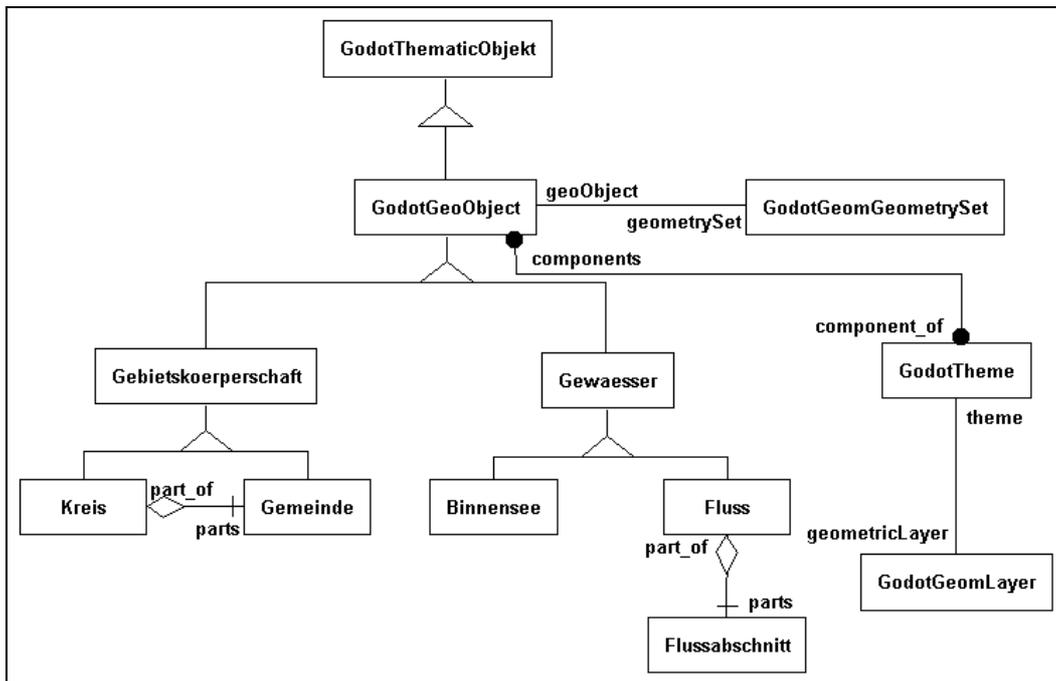


Abb. 1: OMT-Objektdiagramm der Sachobjekte

## Sachobjekte

Die Sachobjekte repräsentieren die thematischen Aspekte einer Anwendung, d.h. die komplexen Sachdaten und die semantischen Beziehungen zwischen den Objekten des Weltausschnitts, der durch die Anwendung abgebildet wird. Für die Sachobjekte existiert eine prinzipiell unvollständige Klassenhierarchie, die für jede Art von Anwendung spezifisch erweitert werden kann (siehe Abb. 1). Die wichtigste Kategorie von Sachobjekten sind die sogenannten Geoobjekte. Gegenüber einem reinen Sachobjekt (z.B. einer Tierart) zeichnet sich ein Geoobjekt (z.B. ein Fluß) durch das Vorhandensein von zugehörigen Geometrieobjekten aus, die seine geographische Lage und Ausdehnung repräsentieren. Da Geoobjekte die wichtigste Art von Sachobjekten in GODOT darstellen, bezieht sich ein Großteil der folgenden Aussagen auf Geoobjekte. Dabei sind viele dieser Aussagen auch auf Sachobjekte verallgemeinerbar, ohne daß dies im Einzelfall besonders hervorgehoben wird.

Geoobjekte können einfach oder komplex strukturiert sein, d.h. sie können aus mehreren anderen Sachobjekten zusammengesetzt sein (z.B. umfaßt ein Landkreis alle seine zugehörigen Gemeinden). Zwischen zusammengesetzten Objekten und ihren Bestandteilen besteht eine 1:n-Beziehung, die durch zwei zueinander inverse Attribute *parts* und *part\_of* realisiert ist. Durch diese generische Partonomiebeziehung, die für die verschiedenen Objektklassen unterschiedlich

typisiert ist, wird in GODOT ein einheitliches Verfahren zum Auffinden der Bestandteil-Objekte bzw. des umfassenden Objekts zur Verfügung gestellt. Darüber hinaus ermöglicht sie die Ableitung von Sachattributen und Geometrien für zusammengesetzte Geoobjekte, aus den entsprechenden Attributen der Bestandteil-Objekte, durch spezifische Aggregationsfunktionen (z.B. entspricht die Einwohnerzahl eines Landkreises der Summe der Einwohner in den zugehörigen Gemeinden, und seine Fläche entspricht der geometrischen Vereinigung der Flächen der einzelnen Gemeinden). Außer dieser vordefinierten Partonomiebeziehung können in Anwendungen auch weitere Partonomien und allgemeine Objektbeziehungen mit besonderer Semantik definiert werden.

Für alle Klassen von Sachobjekten, die zu einer bestimmten Anwendung gehören, werden automatisch Metaobjekte generiert. Diese beschreiben die strukturellen Aspekte der Klassen (z.B. die Oberklassen-Unterklassen-Beziehungen und die Gesamtheit der Attribute einer Objektklasse) und stellen die Funktionen für den Zugriff auf die Attributwerte einzelner Objekte (Instanzen) zur Verfügung. Durch die Metaobjekte wird eine Art Data Dictionary realisiert, welches die interaktiven Komponenten der Benutzeroberfläche (Objekteditoren), die zum Browsen und Editieren von Sachattributen und Objektbeziehungen verwendet werden, mit den notwendigen Informationen über die Struktur der Sachobjekte versorgt.

Die Metainformationen zu den Sachobjektklassen einer Anwendung, die dem Benutzer durch eine eigene Oberflächenkomponente (Schema Browser) verfügbar gemacht werden, sind auch bei der interaktiven Formulierung von Ad-Hoc-Abfragen von großer Bedeutung. Der Benutzer wird dabei ebenfalls durch eine generische Oberflächenkomponente (Query Window) unterstützt, die ihm die klassenspezifischen Sachattribute und Objektbeziehungen mit ihren möglichen Abfrageoperationen vorgibt. Die Abfragen können zur Laufzeit für homogene Mengen von Sachobjekten formuliert werden und liefern in der derzeitigen Version von ObjectStore stets Untermengen der Ausgangsmenge als Ergebnis. In Abfragen können benutzerdefinierte Funktionen, die insbesondere geometrische und topologische Prädikate implementieren, und Pfade, die Abfrageprädikate auf Unterobjekte der Sachobjekte ermöglichen, verwendet werden. Abfragen, die öfter wiederverwendet werden sollen, können als eigenständige Objekte in GODOT gespeichert werden.

Eine Beispielanwendung für GODOT ist auf der Basis von Daten aus dem Amtlichen Topographisch-Kartographischen Informationssystem (ATKIS) (Adv 1989) des Landesvermessungsamts Baden-Württemberg realisiert. Diese Beispielanwendung demonstriert die Erweiterbarkeit des GODOT-Datenmodells zur Unterstützung spezifischer GIS-Anwendungen. Zur Übernahme der ATKIS-Daten ist eine Datenschnittstelle zum SQD-Externformat des Geoinformationssystems SICAD implementiert.

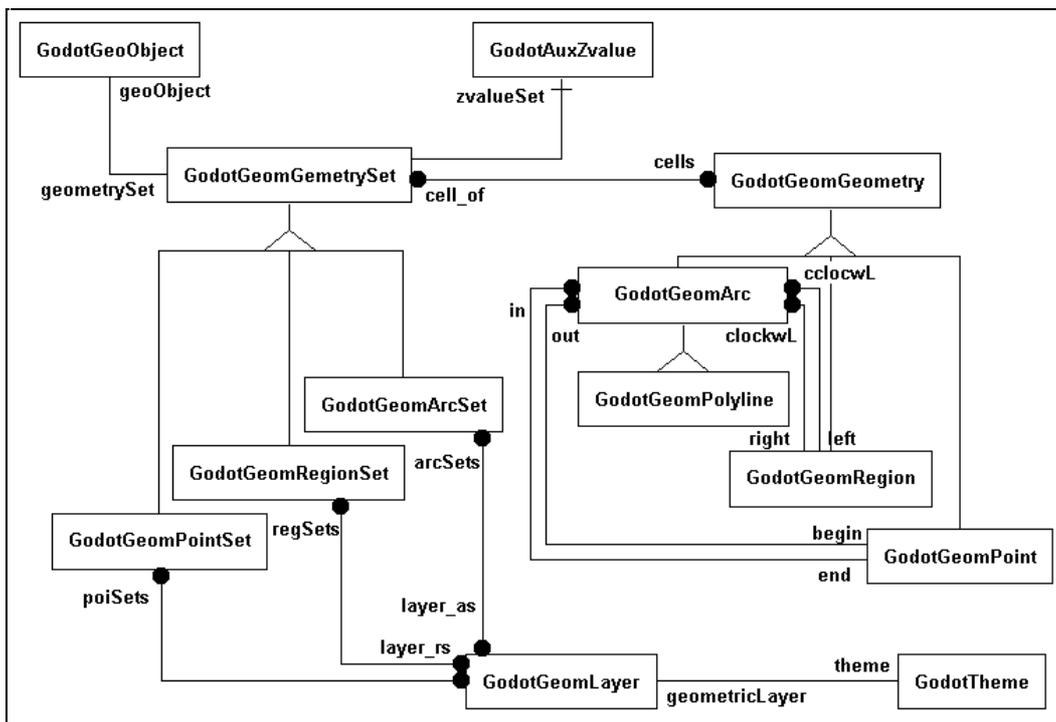


Abb. 2: OMT-Objektdiagramm der Geometrieobjekte

## Geometrieobjekte

Geometrieobjekte dienen zur Repräsentation der Definitionsgeometrien und der topologischen Beziehungen von Geobjekten. Jedem Geobjekt ist eine Menge von geometrischen Elementen zugeordnet. Eine solche Menge, wie auch die einzelnen geometrischen Elemente selbst sind in Form von Datenobjekten repräsentiert. Es gibt drei Arten von geometrischen Elementen, nämlich *Regions* (zweidimensionale, d.h. flächenhafte Elemente), *Arcs* (eindimensionale, d.h. linienhafte Elemente) und *Points* (nulldimensionale, d.h. punktförmige Elemente). Dadurch, daß jedem Geobjekt eine Menge von geometrischen Elementen zugeordnet ist, ist es grundsätzlich möglich, Geobjekte mit *topologisch nicht zusammenhängenden, heterogenen* Geometrien zu definieren. Heterogen heißt in diesem Zusammenhang, daß dem Geobjekt flächenhafte, linienhafte oder punktförmige Geometrien in gemischter Zusammensetzung zugeordnet sind.

Die topologischen Beziehungen zwischen den Geometrieobjekten werden durch Objektverweise explizit gespeichert (siehe Abb 1). Sie ermöglichen es, daß eine Reihe von topologischen Analysen (z.B. Nachbarschaftsanalysen und Netzverfolgung) effizient durchgeführt werden können. Topologische Beziehungen lassen sich in einigen Fällen auf semantische Beziehungen zwischen Geobjekten

(z.B. Zuflüsse zu einem Fluß) abbilden, um dadurch thematische Auswertungen zu unterstützen.

## **Darstellungsobjekte**

In Darstellungsobjekten ist die graphische Erscheinungsform von Sachobjekten festgelegt. Wichtigste Darstellungsobjekte sind die kartographischen Objekte, die der kartographischen Darstellung von Geobjekten dienen. Kartographische Objekte werden persistent in sogenannten Kartographie-Datenbanken gespeichert. Die umfassendsten Objekte in einer solchen kartographischen Datenbank sind die Karten (Instanzen der Klasse Map). Eine Karte umfaßt für einen vorgegebenen Weltausschnitt eine Menge von verschiedenen thematischen Ebenen (Layer), die jeweils die Darstellungsobjekte für zugehörige Mengen von Geobjekten zusammenfassen. Kartographie-Layer können gleichzeitig zu mehreren Karten gehören, was es ermöglicht, Layer mit allgemeiner Bedeutung (z.B. den topographischen Grundriß) in vielen Maps zu verwenden. Darüber hinaus kann eine Map auch Layer mit ähnlichen thematischen Aspekten - aber unterschiedlichem Generalisierungsgrad - beinhalten, die für verschiedene Detaillierungen bei der Darstellung verwendet werden können. Der Kartographie-Layer übernimmt die Funktionen der Erzeugung und Verwaltung der kartographischen Darstellungsobjekte (MapGraphic), die entweder elementar oder komplex strukturiert, d.h. aus anderen MapGraphic-Objekten zusammengesetzt, sein können.

Die kartographischen Darstellungsobjekte verweisen über Referenzobjekte auf ihre zugehörigen Geobjekte, die in separaten Datenbanken abgespeichert sind. MapGraphic-Objekte haben als darstellungsrelevante Attribute außerdem Verweise auf eine eigene Darstellungsgeometrie, die aufgrund von Generalisierungsroutinen erheblich von der Definitionsgeometrie der zugehörigen Geobjekte abweichen kann. Die kartographische Ausgestaltung der MapGraphic-Objekte wird durch Verweise auf Instanzen einer kleinen Zahl von Basissignaturklassen festgelegt, die zur Laufzeit modifiziert werden können.

Die Generierung der MapGraphic-Objekte mit ihren Darstellungsgeometrien und Basissignaturen wird durch Instanzen spezieller Signaturklassen, die eine Methode *createMapGraphic* besitzen, durchgeführt. Diese Methode kann durch komplexe Algorithmen implementiert werden, die auf spezifische Attribute und Funktionen der Geobjekte zurückgreifen können, um die MapGraphic-Objekte für den jeweiligen thematischen Aspekt mit einem bestimmten Generalisierungsgrad zu erzeugen.

Die Auswahl der Geobjekte, für die durch ein komplexes Signaturobjekt jeweils kartographische Darstellungsobjekte erzeugt werden, wird durch eine

benutzerdefinierte Abfrage auf der Menge aller Instanzen der entsprechenden Klasse in einer Geo-Datenbank vorgenommen. Die Assoziationsobjekte (SignatureQuery), die komplexe Signaturobjekte mit den zugehörigen Datenbank-Abfragen verknüpfen werden ebenfalls durch den Kartographie-Layer verwaltet. Dieser kann dadurch seine kartographischen Darstellungsobjekte bei Bedarf jederzeit neu generieren.

Die Bildschirmpräsentation und die Interaktion mit den in den Kartographie-Datenbanken gespeicherten Darstellungsobjekten übernehmen spezialisierte Kartographie-Fenster (MapView), die Teil der Graphikkomponente von GODOT sind. Sie erlauben eine flexible Filterung der dargestellten thematischen und geographischen Informationen durch die Auswahl von Kartographie-Layern in einer Karte, die vom Benutzer interaktiv vorgenommen werden kann. Scroll- und Zoom-Operationen können auf diesen Fenstern durchgeführt werden, ohne die Auswahl der präsentierten Layer zu beeinflussen. Darüber hinaus kann über die sogenannte Detail-Operation der Generalisierungsgrad der dargestellten Thematik dem jeweils aktuellen Maßstab angepaßt werden.

Für die Präsentation der kartographischen Darstellungsobjekte werden in der Graphikkomponente Instanzen der Graphik-Klassenbibliothek GoPATH verwendet, die von Bull France entwickelt wurde. Über diese hierarchisch strukturierten Objekte wird auch die Interaktion mit den zugehörigen Geoobjekten (z.B. die Anzeige der Sachattribute und Objektbeziehungen) gesteuert.

#### **4. Ausblick**

Der GODOT-Prototyp hat als objektorientiertes Geoinformationssystem eine offene Architektur, die anwendungsspezifische Erweiterungen der GODOT-Funktionalität konzeptionell unterstützt. Gleichzeitig ist es möglich, die Funktionalität einzelner Komponenten isoliert für spezielle Anwendungen zu nutzen. Eine solche zunehmend wichtiger werdende Anwendung sind GIS-Datenbankserver, die große Mengen von komplexen raumbezogenen Informationen vorhalten und über eine Anbindung an Computernetze (WAN) einer großen Zahl von potentiellen Nutzern zugänglich machen können. Das GODOT-Kernsystem, das neben den oben beschriebenen Basisklassen und Methoden zur Repräsentation von Sachinformation, Geometrie, Topologie und Kartographie, auch die Funktionalität zur räumlichen und thematischen Clusterung und Indexierung und zur Transaktionsverwaltung zur Verfügung stellt, kann zusammen mit der GODOT-Anfragekomponente die Basis für die Entwicklung eines GIS-Datenbankservers bilden. Durch GIS-Datenbankserver kann die Nutzung großer geographischer Datenbestände (z.B. ATKIS-Daten), deren Erfassung in der Regel mit hohen Kosten verbunden ist, durch eine Vielzahl von Arbeitsgruppen

ermöglicht werden. Bei einer Unterstützung der externen Datenformate verschiedener GIS-Hersteller durch den GIS-Datenbankserver kann gleichzeitig ein wesentlicher Beitrag zur Interoperabilität von GIS geleistet werden.

## **Literatur**

- ACM (1991): Special Section: Next Generation Database Systems, in: Communications of the ACM, Nr. 10, Bd. 34, S. 30-120.
- AdV (1989): Amtliches Topographisch-Kartographisches Informationssystem - ATKIS, Gesamtdokumentation, Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV), Hannover/Bonn.
- Ebbinghaus, J. (1994): Einsatz objektorientierter Datenbanktechnologie im geographischen Informationssystem GODOT, in: Kremers, H. (Hrsg.), Umweltdatenbanken, Reihe "Praxis der Umwelt-Informatik", Bd. 5, Metropolis-Verlag, Marburg, S. 37-58.
- Küspert, K. (1994): Erweiterte relationale Datenbanken: Gestern in der Forschung - heute im Einsatz, in: Kremers, H. (Hrsg.), Umweltdatenbanken, Reihe "Praxis der Umwelt-Informatik", Bd. 5, Metropolis-Verlag, Marburg, S. 9-29.
- Lamb, C./Gordon, L./Orenstein, J./Weinreb, D. (1991): The ObjectStore Database System, in: Communications of the ACM, Nr. 10, Bd. 34, S. 50-63.
- Orenstein, J.A./Manola, F. (1988): Probe: Spatial Data Modelling and Query Processing in an Image Database Application, in: IEEE Transactions on Database Systems, Nr. 5, Bd. 14, S. 611-629.
- Riekert, W.-F. (1993): Das Forschungsprojekt GODOT: Geodatenhaltung mit objektorientierten Techniken, in: 3. Internationales Anwenderforum Geoinformationssysteme, Tagungsband, Duisburg.