

## Geodatenbanksysteme: Gestern, Heute und Morgen

THOMAS BRINKHOFF, Oldenburg

**Keywords:** Spatial Database Systems,  
Data Models, Standardisation,  
Query Processing

**Summary:** *Spatial Database Systems: Past, Present and Future.* During the last decades, spatial database systems have been transformed from a research field known only by few specialists into real-existing commercial systems that form an essential component for building open and interoperable geospatial information infra-structures. In this paper, the essential steps of this progress and the current state-of-the-art are presented. Furthermore, future trends are discussed that are expected to dominate the development in the next years in this field.

**Zusammenfassung:** In den letzten Jahrzehnten haben sich Geodatenbanksysteme von einem Forschungsgebiet weniger Spezialisten zu real existierenden, kommerziell erhältlichen Systemen entwickelt, die einen essenziellen Baustein offener, interoperabler GI-Infrastrukturen bilden. In diesem Beitrag werden die wesentlichen Stationen dieser Entwicklung und der heute erreichte Stand in Wissenschaft und Technik dargestellt. Zudem wird auf die Trends eingegangen, die in den nächsten Jahren diesen Themenbereich voraussichtlich bestimmen werden.

### 1 Einführung

Mit der zunehmenden Nutzung von Geodaten zu kommerziellen, hoheitlichen, wissenschaftlichen und privaten Zwecken steigt auch der Bedarf, raumbezogene Informationen geeignet zu verwalten. Dies ist in erster Linie Aufgabe von *Geoinformationssystemen (GIS)*. Eine wesentliche Teilaufgabe stellt dabei die Speicherung und Abfrage von Geodaten in bzw. aus *Geodatenbanken* dar. Während in der Vergangenheit Geoinformationssysteme die Geodatenbanken eigenständig verwaltet haben, wird diese Aufgabe mehr und mehr separaten Systemen, den *Geodatenbanksystemen* übertragen. Damit erfolgt ein Übergang von proprietären Lösungen zu offenen GI-Infrastrukturen.

In den letzten Jahrzehnten haben sich Geodatenbanksysteme von einem interessanten Forschungsfeld zu real existierenden, kommerziell erhältlichen Systemen entwickelt. Dass dieser Prozess inzwischen weit vorangeschritten ist, wird auch durch die in den letzten Jahren zu diesem Themengebiet entstandenen Lehrbücher von RIGAUX et al. (2002) und BRINKHOFF (2005) verdeutlicht.

Der vorliegende Artikel lässt die wesentlichen Stationen dieser Entwicklung Revue passieren und stellt den heute erreichten Stand in Wissenschaft und Technik dar. Auf dieser Basis soll betrachtet werden, inwiefern Geodatenbanksysteme inzwischen als ausgereifte, praxistaugliche Werkzeuge angesehen werden können und welche weiteren Entwicklungen zu erwarten sind.

### 2 Ausgangssituation

Als das erste GIS der Welt gilt das *Canada Geographic Information System*, welches in den 60er-Jahren in Kanada entwickelt wurde (TOMLINSON 1984). Die Entwickler von CGIS hatten eine Reihe von technischen Problemen zu meistern. So begann die Entwicklung auf einer IBM 1401 mit 16 KB Hauptspeicher, die etwa 1.000 Instruktionen pro Sekunde bewältigte. 1964 kam die IBM 360/65 zum Einsatz, die bei einem Gewicht von 5 Tonnen einen maximalen Hauptspeicher von 512 KB aufwies. Als Hintergrundspeicher wurden zunächst Magnetbänder eingesetzt; Magnetplatten hatten eine viel zu geringe Speicherkapazität, um als Medium für eine Geodatenbank dienen zu können.

Diese Schilderung macht deutlich, dass die Entwicklung von Geoinformationssystemen lange Zeit nicht durch die Verwendung bereits vorhandener Komponenten und Systeme geprägt war, sondern aufgrund technischer und funktionaler Anforderungen Eigenentwicklungen forcierte. So hat auch im Bereich der Datenspeicherung eine Eigenbrötelei bei allen Geoinformationssystemen eine lange Tradition. Bis Ende der 90er-Jahre wurden noch vielfach die Daten ausschließlich in Dateien oder GIS-spezifischen Datenhaltungskomponenten gespeichert. Dies war aufgrund der Schwächen von Standarddatenbanksystemen bei der Verwaltung und Speicherung von Geodaten eine durchaus nachvollziehbare Vorgehensweise.

### 3 Datenmodellierung

Ein *Datenmodell* dient der einheitlichen, formalen Beschreibung von Daten und ihrer Beziehungen. Ein *Datenbankschema* ist das konkrete Datenmodell, das die für einen Anwendungsbereich relevanten Informationen in einer Datenbank beschreibt, wobei das *Datenbankmodell* die Grundsätze festlegt, wie ein Datenmodell in ein Datenbankschema umgesetzt werden kann.

#### 3.1 Relationales Datenbankmodell

Einen entscheidenden Schritt für die heutige Datenbanktechnologie stellte die Entwicklung des *relationalen Datenbankmodells* durch CODD (1970) dar. Anfang der 80er-Jahre kamen *relationale Datenbanksysteme (DBS)* wie IBM DB2, Oracle und Ingres auf den Markt. Heute beträgt der Marktanteil von (objekt-) relationalen Datenbanksystemen rund 80 %. Ebenso hat sich die *Structured Query Language (SQL)* als *Datenbanksprache* allgemein durchgesetzt.

Die Nutzung relationaler Datenbanksysteme zur Verwaltung räumlicher Daten ließ – wie bereits angedeutet – auf sich warten. Kernproblem ist, dass das relationale Datenbankmodell sich auf atomare Datentypen beschränkt, aber geometrische Datenmodelle komplexe Datentypen wie Linienzüge und

Multipolygone erfordern. Die Abbildung solcher Datentypen in das relationale Datenbankmodell bedarf der Verteilung der Daten auf mehrere relationale Tabellen und/oder mehrere Datensätze. Dies ist wenig performant und erschwert oder verhindert die Formulierung und Ausführung von räumlichen Anfragen.

Ein erster Ansatz, relationale Datenbanksysteme im Rahmen von Geoinformationssystemen einzusetzen, stellt die getrennte Speicherung geometrischer und alphanumerischer Attribute dar. Während die nichtgeometrischen Sachattribute in einer relationalen Datenbank gespeichert sind, werden die geometrischen Attribute in einem GIS-spezifischen Datenformat in Dateien abgelegt. Die Kopplung erfolgt meist über einen gemeinsamen Schlüssel. Gegen eine solche Lösung spricht insbesondere, dass die mit einem Einsatz eines DBS verbundenen Vorteile für die geometrischen Attribute verloren gehen. Eine Nutzung der Geodaten außerhalb des spezifischen Geoinformationssystems wird vereitelt, da das Datenformat in der Regel proprietär ist.

Ein heute noch gängiger Ansatz ist die Speicherung von Geometrien in *Binary Large Objects (BLOBs)*. Dies hat eine Reihe von Vorteilen: So kann die gesamte Geometrie in einem Attributwert zusammengefasst werden. Die integrierte Speicherung des gesamten Geoobjektes ist möglich, so dass DBS-Konzepte wie Zugriffskontrolle, gesicherter Mehrbenutzerbetrieb und Recovery auch auf die Geometrien angewendet werden können. Allerdings weist diese Lösung auch essenzielle Nachteile auf: Für das DBS ist ein BLOB eine Folge von Binärinformationen, die es nicht interpretieren kann. Da das DBS die Semantik der Binärdaten nicht kennt, können keine Mechanismen zur effizienten Anfragebearbeitung und -optimierung angewendet werden. Aus den gleichen Gründen ist es auch nicht möglich, geometrische Operationen zu verarbeiten oder durch die Anfragesprache zu unterstützen. BLOBs sind für externe Programme nicht ohne weiteres interpretierbar; ein Anwendungsprogramm ist auf die Nutzung des GIS angewiesen oder muss entsprechend programmierte Komponenten besitzen. Dies widerspricht aber der Idee eines offenen Systems.

### 3.2 Objektrelationales Datenbankmodell

In den 90er-Jahren wurde das Konzept der *Objektorientierung* in einem sehr starken Maße in den Softwareentwurf und in Programmiersprachen eingeführt. Daher lag es nahe, diese Entwicklung auch im Bereich der Datenbankmodelle zu vollziehen. Da sich allerdings die völlig neu konzipierten *objektorientierten Datenbanksysteme* nicht am Markt durchsetzen konnten, ist die Erweiterung des relationalen Datenbankmodells um objektorientierte Konzepte (*objektrelationales Datenbankmodell*) die derzeit relevante Entwicklung (STONEBRAKER & MOORE 1997). Diese Entwicklung hat auch in neue-

ren Versionen von SQL – SQL:1999 und SQL:2003 – ihren Niederschlag gefunden (vgl. TÜRKER 2003).

Das objektrelationale Datenbankmodell ist zur Speicherung von Geodaten gut geeignet: Komplexe geometrische Datentypen können benutzerseitig durch entsprechende Klassen repräsentiert werden. Auch können zur Analyse und Anfragebearbeitung benötigte geometrische Funktionen durch die Programmierung entsprechender Methoden dem DBS hinzugefügt werden.

Ein *objektrelationales Geodatenbanksystem* stellt bereits systemseitig einen hinreichenden Satz von vordefinierten Geometrieklassen und -methoden zur Verfügung. Der Zugriff auf die Geometrieobjekte ist von allen Anwendungsprogrammen aus mit Hilfe von SQL möglich. Damit wird einer wichtigen Anforderung für offene GI-Systeme Rechnung getragen. Da einem objektrelationalen Geodatenbanksystem die Struktur und die Bedeutung der geometrischen Datentypen bekannt sind, wird dieses Wissen auch bei der Anfragebearbeitung berücksichtigt. Derzeit auf dem Markt erhältliche objektrelationale Geodatenbanksysteme sind z.B. Oracle Spatial, IBM Informix mit dem Spatial DataBlade, der IBM DB2 Spatial Extender und PostgreSQL mit PostGIS.

### 3.3 Standardisierung

Einen wichtigen Aspekt offener interoperabler Geoinformationssysteme stellt die *Standardisierung* des verwendeten Geometrieklassenmodells dar. Hier haben das *Open Geospatial Consortium (OGC)* und die *International Organization for Standardization (ISO)* eine Reihe von Spezifikationen erarbeitet.

Grundlage der Standardisierung der Geometrieklassen ist *ISO 19107:2003 „Geographic Information – Spatial Schema“*, das ein konzeptionelles Datenmodell zur Beschreibung der räumlichen Eigenschaften von Geoobjekten beinhaltet. Es umfasst die (Vektor-) Geometrie und Topologie bis zu einer maximalen Dimension von 3 und definiert zugehörige Operationen.

Auf dieser Basis steht das *Simple Feature Model*, das durch *ISO 19125:2004* genormt ist. *Simple Features* sind Geometrien im zweidimensionalen Raum, deren Stützpunkte geradlinig miteinander verbunden sind. Während der erste Teil der Norm das Klassenmodell mit geometrischen Datentypen und zugehörigen Operationen beschreibt (vgl. Abb. 1), behandelt der zweite Teil dessen Umsetzung in ein Datenbankschema.

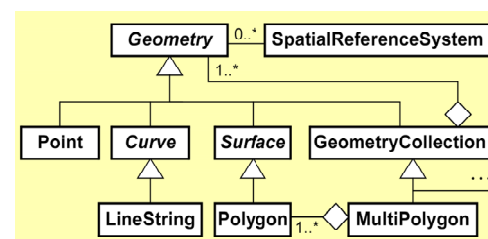


Abb. 1: Teil des Simple Feature Models.

Der ISO/IEC-Standard 13249-3:2003 *SQL/MM Spatial* spezifiziert ebenfalls ein Geometrieklassenmodell. Dieses weist recht große Ähnlichkeiten mit dem Simple Feature Model auf. Wesentliche Erweiterungen von SQL/MM Spatial sind die Unterstützung von Kreisbögen, Methoden zum Validieren von Geometrien, Koordinatentransformationen und Methoden zur Bereitstellung von GML (Geography Markup Language).

### 3.4 Weitere Aspekte

Um in einem Geodatenbanksystem aus den Koordinaten korrekte Längen und Flächen berechnen zu können, ist die Zuordnung und Interpretation eines *räumlichen Bezugssystems* zu einer Geometrie erforderlich. Dementsprechend muss das DBS einen Satz von räumlichen Bezugssystemen zur Unterstützung von geographischen, projizierten und nicht georeferenzierenden Koordinaten bereitstellen. Hierbei haben sich *EPSG-Bezugssysteme* durchgesetzt, die vom „Surveying & Positioning Committee“ der „International Association of Oil & Gas Producers“ (OGP) gepflegt werden.

*Lineare Bezugssysteme* erlauben die Identifizierung von Punkten auf einer Linie durch Abstandsangaben zu einem ausgezeichneten Anfangspunkt. Gerade in Netzwerken – zum Beispiel in Straßennetzen oder in Ver- und Entsorgungsnetzen wie Wasser- und Kanalnetzen – haben lineare Bezugssysteme eine große Bedeutung und werden daher von gängigen Geodatenbanksystemen unterstützt.

*Topologie- und Rasterdatenmodelle* werden bislang von den erwähnten Standards nicht datenbankbezogen abgedeckt, so dass derzeitige Systeme hier entweder keine oder nur eine proprietäre Unterstützung anbieten. Als Beispiel sei hier Oracle Spatial 10 genannt. ORACLE (2005b) enthält ein *Netzwerkdatenmodell*, das u.a. Operationen zur Berechnung kürzester Wege anbietet, und ein *Topologiedatenmodell*, das die Speicherung topologischer Primitive und die Ableitung topologischer Beziehungen aus dem Datenmodell erlaubt. Das *Rasterdatenmodell* von ORACLE (2005a) unterstützt die Speicherung und Abfrage georeferenzierter Rasterkarten mit der Möglichkeit der Kachelung, dem Aufbau von Bildpyramiden und der Kompression der Daten.

## 4 Anfragebearbeitung

Eine Hauptaufgabe eines Geodatenbanksystems ist die Bestimmung von Geoobjekten, deren Geometrie vorgegebene geometrische oder topologische Bedingungen erfüllt. Zur Bearbeitung solcher Anfragen führt das DBS die Anfrage auf eine oder eine Folge von *räumlichen Basisanfragen* zurück. Als Beispiele seien genannt:

- *Punktanfragen*, die zu einem gegebenen Anfragepunkt  $p$  alle Geoobjekte bestimmen, die  $p$  enthalten.
- *Rechteckanfragen*, die zu einem gegebenen Anfragerechteck  $r$  alle Geoobjekte bestimmen, die  $r$  schneiden.
- Der *räumliche Verbund*, der aus zwei Relationen die Paare von Geoobjekten bestimmt, die eine räumliche Verbundbedingung (z.B. ein topologisches Prädikat oder eine Abstandsbedingung) erfüllen. Dies ist Grundvoraussetzung für eine effiziente *Verschneidung* von Karten.
- *Nächste-Nachbarn-Anfragen*, die aus einer Relation die  $k$  nächstgelegenen Geoobjekte in Hinblick auf einen Anfragepunkt  $p$  berechnen.

Die effiziente Bearbeitung räumlicher Basisanfragen wird durch einige Charakteristika von Geodaten erschwert: So ist die Anzahl der Objekte in einer Geodatenbank unter Umständen sehr hoch. Außerdem ist verglichen mit herkömmlichen relationalen Datentypen – Zeichenketten, Zahlen und Datumsangaben – die Verarbeitung geometrischer Datentypen weitaus aufwändiger.

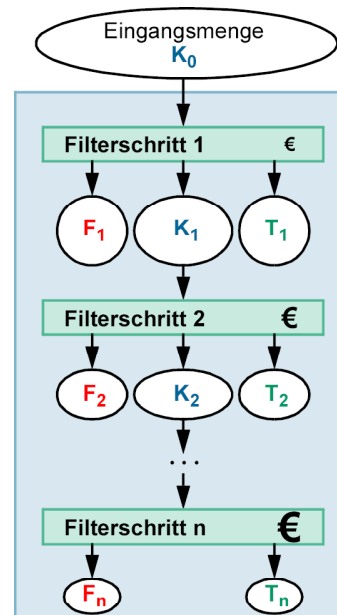


Abb. 2: Mehrstufige Anfragebearbeitung.

### 4.1 Mehrstufige Anfragebearbeitung

Eine Konsequenz aus den genannten Eigenschaften ist die Bearbeitung von räumlichen Anfragen in mehreren *Filterschritten*, wobei der Aufwand pro Objekt mit den Filterschritten zunimmt (siehe Abb. 2). Somit wird versucht, die Menge der Objekte, die die Anfragebedingung potenziell erfüllen können (die *Kandidaten*,  $K_i$ ), in frühen Filterschritten möglichst stark einzuschränken. Kandidaten, bei denen erst später erkannt wird, dass sie die Anfrage nicht erfüllen, heißen *Fehltreffer* ( $F_i$ ). Zusätzliches Ziel dieser Strategie ist es, möglichst viele Objekte, die die Anfrage sicher erfüllen (*Treffer*,  $T_i$ ), durch die Filterschritte zu identifizieren. Im

letzten Filterschritt wird schließlich der aufwändige exakte Test der Anfragebedingung auf einer reduzierten Kandidatenmenge ausgeführt.

Zwei wichtige Formen von Filterschritten stellen *räumliche Indexe* (vgl. Gaede & Günther 1998) und *Approximationen der Geometrie* (Brinkhoff et al. 1993) dar, wobei die gängigste Approximation zu den Koordinatenachsen orthogonale *minimal umgebende Rechtecke* (MURs) sind.

#### 4.2 Räumliche Indexe

In Datenbanksystemen dienen *Indexe* dazu, die Daten so zu organisieren, dass Anfragen bezüglich eines oder mehrerer Anfrageattribute effizient die gesuchten Daten finden. Solchen Indexen liegt typischerweise eine Datenstruktur zugrunde, die eine *lineare Ordnung* benötigt, um die Daten sortiert anordnen zu können. Eine solche lineare Ordnung ist für Zahlen oder Zeichenketten offenkundig, für geometrische Datentypen hingegen nicht. Daher können herkömmliche Indexe nicht (ohne weiteres) für Geodaten eingesetzt werden.

*Räumliche Indexe* müssen in der Lage sein, Approximationen von Geometrien zu verwalten. Dazu zerlegen sie den Datenraum dynamisch in kleinere räumliche Einheiten, den *Blockregionen*. Um räumliche Basisanfragen effizient ausführen zu können, sollten Daten, die räumlich benachbart sind und daher von räumlichen Anfragen häufig gemeinsam benötigt werden, mit hoher Wahrscheinlichkeit zusammen in einer Blockregion abgespeichert werden.

In den 80er-Jahren wurden eine Reihe von *Punktstrukturen* entwickelt, die in der Lage sind, effizient zwei- oder mehrdimensionale Punkte zu organisieren (WIDMAYER 1991). Für die Speicherung von Geometrien und Approximationen, die eine Ausdehnung besitzen, sind Punktstrukturen – zunächst – nicht geeignet. Daher wurden Verfahren entwickelt, die in der Lage sind, linien- und flächenhafte Geometrien zu organisieren (SEEGER & KRIEGL 1988).

Das *Clipping* ordnet eine Geometrie jeder Blockregion zu, die sie schneidet. Dadurch können Duplikate entstehen und die Anzahl der Einträge im Index stark ansteigen.

*Punkttransformationen* überführen  $k$ -dimensionale Rechtecke in  $2k$ -dimensionale Punkte, so dass Punktstrukturen als Index verwendet werden können. Da aber die geometrischen Verhältnisse aufgrund der Einbeziehung der Ausdehnung verloren gehen, bleibt die räumliche Nachbarschaft von Objekten nicht hinreichend erhalten.

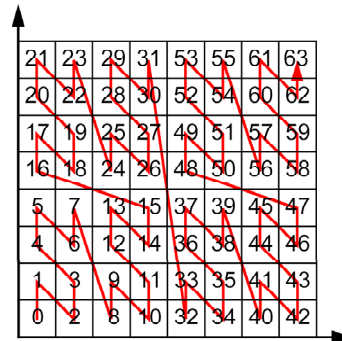


Abb. 3: z-Ordnung.

*Fraktale Kurven* erlauben eine Einbettung eines mehrdimensionalen Raums in den eindimensionalen Raum. Dazu wird der Datenraum in Zellen gleicher Größe eingeteilt und diese Zellen über die fraktale Kurve durchnummeriert. Damit entsteht eine lineare Ordnung zwischen den Zellen. Abb. 3 zeigt dies anhand der *z-Ordnung*. Somit kann ein herkömmlicher Index zur Organisation der Geodaten verwendet werden. Der *relationale lineare Quadtree* (WANG 1991) beruht auf diesem Ansatz. Allerdings sind fraktale Kurven nicht in der Lage, an allen Stellen des Datenraums räumliche Nähe zu erhalten.

Das Prinzip der *überlappenden Blockregionen* erlaubt eine nicht disjunkte Aufteilung des Datenraums. Durch die Überlappungen der Blockregionen kann ein willkürliches Zerschneiden von Geometrien vermieden werden. Der *R-Baum* von GUTTMAN (1984) beruht auf diesem Ansatz. Hauptschwierigkeit bei der Technik überlappender Blockregionen ist es, die Überlappungen möglichst klein zu halten. Durch die Überlappung wird die Effizienz räumlicher Anfragen negativ beeinflusst, da die Verarbeitung dort in mehrere Teilbäume verzweigen muss. Daher haben späterer Varianten des R-Baums versucht, die Überlappungen zu verringern. Hier sei insbesondere der *R\*-Baum* genannt (BECKMANN et al. 1990).

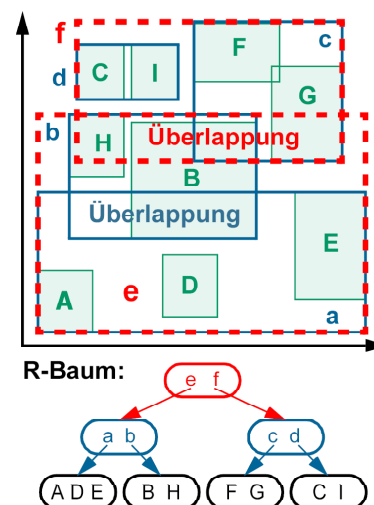


Abb. 4: Überlappende Blockregionen und R-Baum.

### 4.3 Komplexe Basisanfragen

Während die Bearbeitung einfacher räumlicher Basisanfragen wie Punkt- und Rechteckanfragen sich unmittelbar aus der Datenstruktur des räumlichen Indexes ableitet, bedürfen komplexere Basisanfragen wohlüberlegter Verfahren. Als Beispiele hierfür sind der *räumliche Verbund* zwischen R-Bäumen (BRINKHOFF et al. 1993) und die Bearbeitung von *Nächste-Nachbarn-Anfragen* (HJALTASON & SAMET 1999).

## 5 Künftige Entwicklung

In den vorhergehenden Kapiteln wurde der Stand wiedergegeben, den Geodatenbanksysteme in den letzten Jahrzehnten erreicht haben. Nun sollen aktuelle Trends betrachtet werden.

### 5.1 3D-Geodatenbanken

In den letzten Jahren haben *3D-Geoinformationssysteme* deutlich an Bedeutung gewonnen (COORS & ZIPF 2005). Dies legt die Verwendung von *3D-Geodatenbanksystemen* nahe. Allerdings benötigt man für die Speicherung von 3D-Geodaten nicht immer ein 3D-Geodatenbanksystem. Oftmals reicht es, die Höheninformation als zusätzliches thematisches Attribut zu speichern (2,5D-Datenmodell). Der Bedarf, ein 3D-Geodatenbanksystem einzusetzen, ist davon abhängig, ob die dritte Dimension hinsichtlich Ausdehnung und Verwendung in Anfragebedingungen gleichberechtigt zu den ersten beiden Dimensionen ist. Auch ist zu betrachten, ob dreidimensionale Funktionen (z.B. die Berechnung des Schnitts zwischen zwei Körpern) benötigt werden.

Ein 3D-Geodatenbanksystem muss dem Anwender ein geeignetes Datenbankmodell bereitstellen und die für 3D-Anwendungen erforderlichen räumlichen Basisanfragen durch entsprechende Indexe und geometrische Algorithmen unterstützen.

ISO 19107 „Spatial Schema“ enthält sowohl geometrische als auch topologische 3D-Primitive, wobei allerdings nur eine vektorbasierte Randrepräsentation vorliegt. Die datenbankbezogenen ISO- und OGC-Spezifikationen berücksichtigen 3D-Geometrien hingegen bislang nur sehr rudimentär. Gleiches gilt für die derzeitigen objektrelationalen Geodatenbanksysteme. Hier sind proprietäre Erweiterungen zu erwarten. Solche Datenmodelle müssen analog zum Simple Feature Model geometrische Funktionen anbieten, die Körper als Eingangsparameter und ggf. auch als Resultat haben.

### 5.2 Verwaltung von Messdaten

Mit dem Aufkommen moderner Messverfahren wie dem Laserscanning steigen die Datenvolumina georeferenzierter Messdaten massiv an. Dadurch entstehen sehr große Geodatenbanken im Terabyte-Bereich. Ein einzelnes Messdatum hat dabei eine recht einfache Struktur: typischerweise eine (3D-)

Punktcoordinate mit einem Messwert oder einem Vektor von Messwerten.

Messsysteme verwalten Messdaten meist dateigestützt und überlassen die Auswertung speziellen Programmen oder CAD-Erweiterungen. Aufgrund der Datenvolumina stößt dieser Ansatz aber an seine Grenzen. Derzeitige Geodatenbanksysteme sind hingegen auf die Verwaltung bereits aufbereiteter geographischer Daten ausgerichtet. Somit besteht die Anforderung, die Kluft zwischen diesen beiden Ansätzen zu schließen und Anfragen, die zur Analyse der Messdaten erforderlich sind, durch Geodatenbanksysteme zu unterstützen. Als Beispiel sei hier die Bereitstellung von Basisanfragen genannt, die zur Extraktion und Approximation von Flächen und Kanten aus Punktwolken benötigt werden (BRINKHOFF 2004).

### 5.3 Spatio-temporale Datenbanken

Bei Geodaten spielen insbesondere das Miteinander von Raum und Zeit eine entscheidende Rolle. So möchte man zum Beispiel

- wissen, wann das Römische Reich seine größte Ausdehnung besaß,
- alle Fahrzeuge abfragen, die schneller als 120 km/h fahren oder
- alle Personen bestimmen, die voraussichtlich in den nächsten 15 Minuten eine Filiale passieren werden.

Bei allen drei Beispielen liegt eine Kombination zwischen räumlichen und zeitlichen Anfragekriterien vor. Dafür bieten aber derzeitige Geodatenbanksysteme keine geeigneten Operationen und Indexe an. Dies hat in den letzten Jahren zur Entwicklung von *spatio-temporalen Datenbankmodellen* und *Datenstrukturen* geführt – eine Entwicklung, die zurzeit noch keinen Abschluss gefunden hat. Für eine ausführliche Darstellung zu dieser Thematik sei hier auf die Darstellung von GÜTING & SCHNEIDER (2005) verwiesen.

### 5.4 Sensornetzwerke

Die technologische Entwicklung ermöglicht es zunehmend, Sensoren, Prozessoren und autarke Energiequellen preisgünstig auf kleinstem Raum zu integrieren. Über drahtlose Kommunikationseinrichtungen können solche *Sensorknoten* ein (drahtloses) *Sensornetzwerk* bilden, in dem die Knoten miteinander kommunizieren können. Die Aufgabe eines Sensorknotens besteht dann u.a. im Auslesen der Messdaten von den Sensoren und der Weiterleitung dieser Daten. Sensorknoten haben eine Lage im Raum, die für eine korrekte Interpretation der Messwerte oftmals von hohem Interesse ist. Damit produziert ein Sensorknoten Geodaten.

Auf dieser Basis kann man den Ansatz verfolgen, die Sensorknoten mit einfacher DBS-Funktionalität auszustatten (WOO et al. 2004). Dies ermöglicht einem Client, eine Anfrage an das Sensornetzwerk

zu stellen, die dann von den einzelnen Sensorknoten autonom weitergeleitet und verarbeitet wird, so dass ein Anfrageergebnis zurück übermittelt werden kann. Dabei sind prinzipiell auch räumliche Anfragen vorstellbar; das Sensornetzwerk ist dann wie in Abb. 5 Teil eines Geodatenbanksystems.

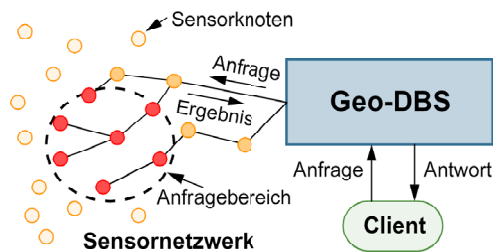


Abb. 5: Sensornetzwerk als Geodatenbank.

## 6 Schlussbetrachtung

Ein solcher Artikel kann naturgemäß nur einen groben Überblick über die Entwicklung von Geodatenbanksystemen in den letzten Jahrzehnten geben. Viele Aspekte und Teilgebiete konnten nur kurz angerissen werden oder wurden – wie z.B. spezifische geometrische Algorithmen – gar nicht erwähnt. Der interessierte Leser sei hier auf die aufgeführte Literatur verwiesen. Die nachfolgenden Aussagen sollen ein kurzes Fazit darstellen:

- Objektrelationale Geodatenbanksysteme stellen ein essenzielles und ausgereiftes Werkzeug zur Speicherung und Abfrage von (zweidimensionalen) Geodaten dar.
- Sie sind ein wesentlicher Baustein für offene GI-Systeme, die es erlauben, Geodaten interoperabel in eine DV-Infrastruktur zu integrieren.
- Die weitere Entwicklung wird das Einsatzfeld von Geodatenbanksystemen weiter vergrößern.

### Literatur

- BECKMANN, N., KRIEGEL, H.-P., SCHNEIDER, R., SEEGER, B., 1990: The R\*-tree: An Efficient and Robust Access Method for Points and Rectangles. – Proceedings ACM SIGMOD International Conference on Management of Data, Atlantic City: 322-331.
- BRINKHOFF, T., 2004: Spatial Access Methods for Organizing Laserscanner Data. – Proceedings 20<sup>th</sup> Congress of the International Society of Photogrammetry and Remote Sensing, Istanbul.
- BRINKHOFF, T., 2005: Geodatenbanksysteme in Theorie und Praxis. – 466 S., Wichmann Verlag, Heidelberg.
- BRINKHOFF, T., KRIEGEL, H.-P., SCHNEIDER, R., 1993: Comparison of Approximations of Complex Objects used for Approximation-based Query Processing in Spatial Database Systems. – Proceedings 9<sup>th</sup> International Conference on Data Engineering, Wien: 40-49.
- BRINKHOFF, T., KRIEGEL, H.-P., SEEGER, B., 1993: Efficient Processing of Spatial Joins Using R-trees. – Proceedings ACM SIGMOD International Conference on Management of Data, Washington DC: 237-246.
- CODD, E. F., 1970: A Relational Model for Large Shared Data Banks. – Communications of the ACM, 13(6): 377-87.

- COORS, V., ZIPF, A. (Hrsg.), 2005: 3D-Geoinformationssysteme. – 522 S., Wichmann Verlag, Heidelberg.
- GAEDE, V., GÜNTHER, O., 1998: Multidimensional Access Methods. – ACM Computing Surveys, 30(2): 170-231.
- GÜTING, R. H., SCHNEIDER, M., 2005: Moving Objects Databases. – 389 S., Morgan Kaufmann Publishers, San Francisco.
- GUTTMAN A., 1984: R-trees: A Dynamic Index Structure for Spatial Searching. – Proceedings ACM SIGMOD International Conference on Management of Data, Boston: 47-57.
- HJALTASON, G. R., SAMET, H., 1999: Distance Browsing in Spatial Databases. – ACM Transactions on Database Systems, 24(2): 265-318
- ORACLE CORP., 2005a: Oracle Spatial – GeoRaster, 10g Release 2.
- ORACLE CORP., 2005b: Oracle Spatial – Topology and Network Data Models, 10g Release 2.
- RIGAUX, P., SCHOLL, M., VOISARD, A., 2002: Spatial Databases, With Applications to GIS. – 410 S., Morgan Kaufmann Publishers, San Francisco.
- SEEGER B., KRIEGEL, H.-P., 1988: Techniques for Design and Implementation of Efficient Spatial Access Methods. – Proceedings 14<sup>th</sup> International Conference on Very Large Data Bases, Los Angeles: 360-371.
- STONEBRAKER, M., MOORE, D., 1997: Object-relational DBMSs: the next great wave. – 216 S., Morgan Kaufmann Publishers, San Francisco.
- TOMLINSON, R. F., 1984: Geographic Information Systems – A New Frontier. – The Operational Geographer/La Géographie Appliquée, 1984(5): 31-36.
- TÜRKER, C., 2003: SQL:1999 & SQL:2003. – 478 S., dpunkt.verlag, Heidelberg.
- WANG, F., 1991: Relational-Linear Quadtree Approach for Two-Dimensional Spatial Representation and Manipulation. – IEEE Transactions on Knowledge and Data Engineering, 3(1): 118-122.
- WIDMAYER, P., 1991: Datenstrukturen für Geodatenbanken. – In: VOSSEN, G., WITT, K.-U. (Hrsg.): Entwicklungstendenzen bei Datenbank-Systemen. – Oldenbourg Verlag, München: 317-361.
- WOO, A., MADDEN, S., GOVINDAN, R., 2004: Networking Support for Query Processing in Sensor Networks. – Communications of the ACM, 47(6): 47-52.

Anschrift des Autors:

Prof. Dr. THOMAS BRINKHOFF  
 Institut für Angewandte Photogrammetrie und Geoinformatik (IAPG)  
 FH Oldenburg/Ostfriesland/Wilhelmshaven  
 Ofener Str. 16, D-26121 Oldenburg  
 e-mail: thomas.brinkhoff@fh-oldenburg.de

Manuskript eingereicht: Mai 2006

Angenommen: Juni 2006