

Realisierung eines mobilen Geodatenservers für PDAs auf der Basis von Standards des OpenGIS-Consortiums

Stefan Schmitz

Alexander Zipf

Hidir Aras

European Media Laboratory EML
Heidelberg

<Vorname>.<Nachname>@eml.villa-bosch.de

28. Januar 2002

1 Problemstellung

1.1 Das Deep Map System

Deep Map, ein Leitprojekt des European Media Laboratory (EML), hat die Entwicklung eines tragbaren Systems, beispielsweise auf Basis eines *Personal Digital Assistant (PDA)*, zum Ziel, welches als intelligenter elektronischer Touristenführer in Heidelberg eingesetzt werden kann. Langfristig soll das Deep Map System in der Lage sein, personalisierte Stadtrundgänge durch Heidelberg zu generieren und Touristen bei der Erkundung in der Stadt zu unterstützen. Bei der Planung eines Stadtrundgangs finden verschiedene Einflussgrößen Berücksichtigung, beispielsweise persönliche Interessen, Bedürfnisse, sowie der kulturelle und soziale Hintergrund des Benutzers (vgl. Malaka und Zipf 2000). Ein erster Prototyp wurde bereits in Zipf et al. (2000) vorgestellt.

Im Rahmen des verteilten Geoinformationssystems von Deep Map wird derzeit ein Geodatenbanksystem mit zentralen Servern eingesetzt (vgl. Zipf und Aras 2001). Der mobile Client bietet hier noch keinerlei Funktionalität bezüglich der Verarbeitung oder Speicherung von Geodaten (GIS-Funktionalität). Vielmehr leitet das mobile Gerät alle anfallenden Anfragen über drahtlose Kommunikationswege (z. B. Wireless LAN) an den Server weiter. Der Client dient lediglich zur Visualisierung und Interaktion mit dem Benutzer. Das verwendete Geodatenbanksystem (Spatial Database Engine (SDE) von ESRI) ist für den Einsatz auf mobilen Geräten aufgrund der benötigten Netzwerk-, Speicher- und Rechenkapazitäten ungeeignet. Mit dem hier vorgestellten, neu entwickelten Geodatenadapter soll GIS-Funktionalität auf den mobilen Client verlagert werden, die bisher i. d. R. nur serverbasierte Lösungen anbieten konnten. Gerade durch die aktuelle Entwicklung von Location Based Services (LBS) für Dienste der kommenden Mobilfunkgenerationen ist dies von Bedeutung (vgl. Zipf 2001). Nun können neuartige Anwendungen realisiert werden, die auch Ressourcen des Client für GIS-Dienste nutzen und damit die Abhängigkeit von Netzwerkverfügbarkeit verringert.

Dazu müssen eine Reihe von Anforderungen erfüllt werden. Um auf Daten unabhängig von einem Server zugreifen zu können, müssen diese lokal vorliegen, also lokal persistent gespeichert werden. Abfragen an die lokal vorliegende Datenbasis sollten effizient beantwortet werden können. Zur Unterstützung einer effizienten Abfragebearbeitung müssen geeignete Zugriffsmechanismen zur Verfügung gestellt werden.

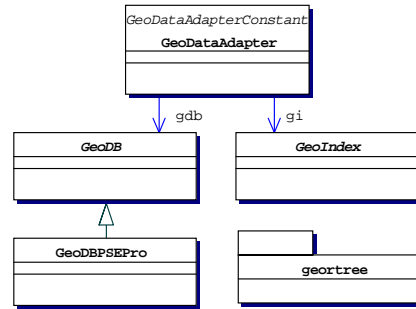


Abbildung 1: Architektur des Geodatenadapters

1.2 Zielplattform

Der Geodatenadapter soll möglichst plattformunabhängig implementiert werden, damit ein flexibler Einsatz möglich ist. Zielplattformen für den Prototypen sind Personalcomputer mit verschiedenen Betriebssystemen und der Pocket PC iPAQ H 3630 von Compaq mit einer Festplatte (IBM Microdrive 1GB). Um eine weitgehende Plattformunabhängigkeit zu erreichen wird der Prototyp des Geodatenadapters in Java implementiert. So kann der Geodatenadapter prinzipiell auf allen Systemen eingesetzt werden, die eine Java 2 Umgebung zur Verfügung stellen, so auch auf Personalcomputern und dem Compaq iPAQ. Einschränkend wirken nur die Eigenschaften des verwendeten Systems, beispielsweise die Größe des Speichers oder die Geschwindigkeit des Prozessors. Eine Java 2 Umgebung für den iPAQ wird von Blackdown (<http://www.blackdown.org>) angeboten. Dazu muss auf dem Gerät Linux (z. B. Familiar-Linux, <http://familiar.handhelds.org>) als Betriebssystem vorliegen.

1.3 Abfragen

Durch den Geodatenadapter sollen thematische, räumliche und gemischte Abfragen bearbeitet werden können. Thematische Abfragen sind Abfragen, die sich auf nicht-räumliche Attribute eines Objekts beziehen (z. B. Hausnummern, Namen). Räumliche Abfragen dagegen beziehen sich auf Attribute, die die räumliche Lage und Ausdehnung eines Objekts beschreiben. Eine Straße oder ein Fluss können beispielsweise durch einen Linienzug beschrieben werden, dessen Lage im Raum durch die Koordinaten der seiner Punkte festgeschrieben ist. Eine räumliche Abfrage ist z. B. „Finde alle Straßen, die den gegebenen Fluss schneiden“.

2 Der Geodatenadapter

2.1 Architektur des Geodatenadapters

Die Anforderungen der persistenten Speicherung von Geobjekten und der effizienten Bearbeitung von geometrischen und thematischen Abfragen werden durch ein räumliches Datenbanksystem erfüllt. Ein räumliches Datenbanksystem ist ein Datenbanksystem, welches in seinem Datenmodell, seiner Abfragesprache und seiner Implementierung räumliche Datentypen und Operationen unterstützt (vgl. Güting 1994). Rigaux et al. (2002) gibt eine aktuelle Übersicht über kommerzielle Geoinformationssysteme und Geodatenbanksysteme. Diesen ist gemein, dass sie nicht auf PDAs wie dem Pocket PC iPAQ ausgeführt werden können. Zur Realisierung des Geodatenadapters muss daher die Funktionalität eines räumliches Datenbanksystems implementiert werden. Bei der Entwicklung des Geodatenadapters wird der Entwicklungsaufwand verringert, indem ein nicht-räumliches Datenbanksystem zur Verwaltung und Speicherung von

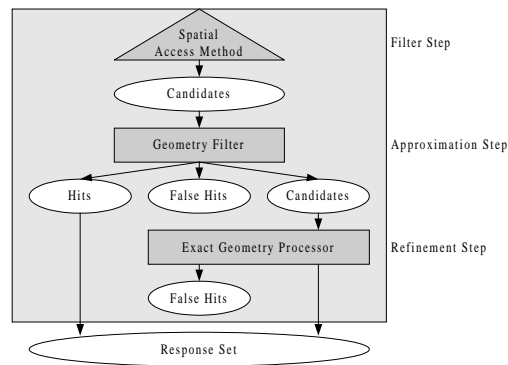


Abbildung 2: Mehrstufige Abfragebearbeitung (vgl. Brinkhoff et al. 1994)

Objekten und zur Bearbeitung thematischer Abfragen eingesetzt wird. Neben dem Datenbanksystem findet ein räumlicher Index zur Verwaltung der Objekte anhand ihrer räumlichen Lage und Ausdehnung Verwendung. Während im Datenbanksystem die Objekte selbst gespeichert werden, verwaltet der räumliche Index lediglich Referenzen auf Objekte. Der Geodatenadapter vereinigt Datenbanksystem und räumlichen Index und stellt die Funktionalität eines räumlichen Datenbanksystems nach außen zur Verfügung. Datenbanksystem und räumlicher Index werden über das Adapter-Entwurfsmuster (vgl. Adapter Pattern, Gamma et al. 1995) und die abstrakten Klassen `GeoDB` und `GeoIndex` eingebunden. So ist der Geodatenadapter nicht auf die Verwendung eines bestimmten Datenbanksystems oder eines bestimmten räumlichen Index festgelegt. Abbildung 1 zeigt die Architektur des Geodatenadapters. Die Klasse `GeoDataAdapter` integriert das Datenbanksystem und den räumlichen Index. Über die Adapterklasse `GeoDBPSEPro` wird das Datenbanksystem *PSE Pro for Java* (<http://www.objectdesign.com>) eingebunden. Als räumlicher Index wird ein R-Tree (vgl. Guttman 1984) verwendet, der sich im Paket `geortree` befindet.

2.2 Datenmodell

Die Geography Markup Language (GML) ist eine XML-Kodierung für den Transport und die Speicherung geographischer Informationen. Sie stellt einen offenen, herstellerunabhängigen Rahmen zur Definition von räumlichen Anwendungen und Objekten dar. Es werden Basistypen bereitgestellt, die von Anwendungen zur Deklaration eigener Typen benutzt werden können (vgl. Open GIS Consortium 2001). Als Datenmodell wird das GML Feature Schema und GML Geometry Schema des Open GIS Consortium verwendet. Die Geometrien räumlicher Objekte werden durch Vektordaten des GML Geometry Schemas repräsentiert. Aus den XML-Schemata werden Java-Klassen abgeleitet. Diese Klassen dienen als Basistypen der im Geodatenadapter verwalteten Objekte. Import und Export von Daten kann über GML erfolgen.

2.3 Abfragebearbeitung

Die räumlichen Attribute von Geobjekten weisen eine hohe Komplexität durch der Anzahl der Punkte bzw. Kanten, ihrer Ausdehnung, ihrer Form und ihrer Verteilung im Raum auf (vgl. Brinkhoff et al. 1994). Das Ausführen räumlicher Operationen ist aufgrund dieser Komplexität rechenintensiv, z. B. die Berechnung der Schnittmenge zweier Polygone. Das Bearbeiten einer räumlichen Abfrage durch einfaches sequentielles Überprüfen aller Objekte der Datenbasis ist deshalb nicht praktikabel. Um die Anzahl der zu überprüfenden Objekte einzugrenzen, erfolgt die Bearbeitung einer räumlichen Abfrage in mehreren Stufen (vgl. Brinkhoff et al. 1994). Das schrittweise Vorgehen wird in Abbildung 2 verdeutlicht.

Im **Filter Step** wird mithilfe eines räumlichen Index der Datenraum der Abfrage eingegrenzt. Zur Organisation der Objekte in einem räumlichen Index werden Approximationen verwendet. So können schnelle und einfache geometrische Operationen verwendet werden. Es wird eine Kandidatenmenge ermittelt, die in den weiteren Schritten genauer untersucht wird.

Im **Approximation Step** erfolgt eine genauere Untersuchung der Kandidaten aus Schritt 1, ohne jedoch die genaue Geometrie der Objekte zu betrachten. Als Ergebnis dieses Prozesses entstehen drei Mengen: Die Menge der Treffer (Hits), also der Objekte, die den Bedingungen der Abfrage genügen; die Menge der falschen Treffer (False Hits), die bei näherer Betrachtung nicht mehr den Abfragebedingungen genügen; alle übrigen Objekte verbleiben in der Kandidatenmenge.

In der Kandidatenmenge befinden sich nun nur noch Objekte, für die durch Verwendung von Approximationen in den vorgelagerten Bearbeitungsschritten nicht festgestellt werden kann, ob sie den Abfragebedingungen genügen. Für diese Objekte müssen im **Refinement Step** die exakten Geometrien betrachtet werden. Nachdem ein Objekt als Teil der Ergebnismenge identifiziert wurde, kann es an weitere Bearbeitungsstufen weitergegeben oder einfach der Ergebnismenge hinzugefügt werden.

Thematische Abfragen werden direkt an das Datenbanksystem weitergeleitet. Die Abfragesprache ist allein vom verwendeten Datenbanksystem abhängig. Das Datenbanksystem bearbeitet die thematische Abfrage mithilfe seiner internen Abfragebearbeitungs- und Indizierungsmechanismen. Das Ergebnis wird an den Geodatenadapter zurückgeliefert und von dort an den Abfragesteller weitergegeben.

Die Bearbeitung **gemischter Abfragen** erfolgt in zwei Stufen. Zuerst wird der räumliche Abfrageteil bearbeitet. Die Lösungsmenge der räumlichen Abfrage wird dem Datenbanksystem zusammen mit dem thematischen Abfrageausdruck zur Bearbeitung übergeben. Als Ergebnis sind alle Objekte der Datenbasis bestimmt, die sowohl den räumlichen als auch den thematischen Abfragebedingungen genügen.

3 Zusammenfassung und Ausblick

Es wurde der Prototyp eines mobilen GeodatenServers vorgestellt, der folgende Eigenschaften realisiert: Räumliche Objekte mit vektoriellen Geometrien werden im Geodatenadapter persistent gespeichert. Dazu wird ein Datenbanksystem über eine Adapterklasse in den Geodatenadapter eingebunden. Thematische und räumliche Abfragen können bearbeitet werden. Thematische Abfragen werden an das eingebundene Datenbanksystem weitergeleitet. Für den Prototypen wird das Datenbanksystem *PSE Pro for Java* verwendet. Die effiziente Bearbeitung räumlicher Abfragen wird durch die Integration eines räumlichen Index unterstützt. Der räumliche Index wird, wie das Datenbanksystem, über eine Adapterklasse in den Geodatenadapter eingebunden. Im Prototypen steht ein R-Tree als räumlicher Index zur Verfügung.

Der Geodatenadapter ist vollständig in Java implementiert. Auch das Datenbanksystem *PSE Pro for Java* benötigt lediglich eine Java 2 kompatible Umgebung. Damit ist der Geodatenadapter plattformunabhängig realisiert und kann auf allen Systemen, die Java 2 unterstützen, eingesetzt werden. Durch die vollständige Implementierung in Java kann der Geodatenadapter problemlos in andere Java-Anwendungen integriert werden.

Testläufe unter Verwendung einer Datenbasis von 1899 Objekten mit jeweils einem Polygon als räumlichem Attribut wurden erfolgreich durchgeführt. Dabei zeigte sich, dass nicht die Rechenleistung oder Hauptspeichergröße, sondern die Zugriffsgeschwindigkeit des Microdrives die Performance des Geodatenadapters bestimmt.

Durch den Geodatenadapter kann GIS-Funktionalität auf den mobilen Client eines verteilten Geoinformationssystems, wie es in Deep Map realisiert ist, verlegt werden. So wird die Ausfallsicherheit des Systems erhöht, da Daten redundant auf dem Server und dem Client vorliegen. Die Verfügbarkeit des Clients wird erhöht. Funklöcher führen nicht dazu, dass keine Abfragen mehr

vorgenommen werden können. Vielmehr kann mit den lokal vorhandenen Daten weitergearbeitet werden.

Die auf dem Client verfügbare Speicherfunktionalität ermöglicht die zukünftige Realisierung intelligenter Caching-Strategien in Verbindung mit Ressourcenadaptivität. Beispielsweise kann ein mobiler Client erkennen, in welche Richtung sich der Anwender mit dem Client bewegt. Bei guter Netzwerkverbindung können dann im Voraus Daten vom Server abgefragt und lokal gespeichert werden, die möglicherweise im weiteren Verlauf der Anwendung benötigt werden. Falls zu einem späteren Zeitpunkt die Netzwerkqualität nicht mehr ausreichend sein sollte, kann mit dem auf dem Client vorhandenen Datenbestand weitergearbeitet werden.

Literatur

- Brinkhoff, T., Kriegel, H.-P., Schneider, R., Seeger, B.** *Mult-Step Processing of Spatial Joins*; in: Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, Minneapolis, Minnesota, May 24 – 27, 1994, ACM Press, S.197–208
- Gamma, E., Helm, R., Johnson, R., Vlissides, J.** *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Publishing Company, Reading u. a., 1995
- Güting, R. H.** *An Introduction to Spatial Database Systems*; in: VLDB Journal, Volume 3, Number 4, October 1994, S.357–399
- Guttman, A.** *R-Trees: A Dynamic Index Structure for Spatial Searching*; in: SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, Beatrice Yormark (Eds.), ACM Press, 1984, S.47–57
- Malaka, R. und Zipf, A.** *DEEP MAP – Challenging IT research in the framework of a tourist information system*; in: Fesenmaier, D. R., Klein, S. und Buhalis, D. (Eds.): Information and Communication Technologies in Tourism 2000, Proceedings of ENTER 2000, 7th International Congress on Tourism and Communications Technologies in Tourism, Barcelona, Spain, Springer Computer Science, Wien, New York, 2000, S.15–27
- Open GIS Consortium** *Geography Markup Language (GML) 2.0*, 2001, <http://www.opengis.net/gml/01-029/GML2.html> (28.12.2001)
- Rigaux, P., Scholl, M. und Voisard, A.** *Spatial Databases: With Application to GIS*, Morgan Kaufmann Publishers, San Francisco u. a., 2002
- Zipf, A.** *Interoperable GIS-Infrastruktur für Location-Based Services (LBS) – M-Commerce und GIS im Spannungsfeld zwischen Standardisierung und Forschung*; in: GIS Geo-Informationssysteme, Zeitschrift für raumbezogene Information und Entscheidungen, 09/2001, S.37–43
- Zipf, A. und Aras, H.** *Realisierung verteilter Geodatenserver mit der OpenGIS SFS für CORBA*; in: GEO-INFORMATIONSYSTEME, Zeitschrift für raumbezogene Information und Entscheidungen, Herbert Wichmann Verlag, Hüthig GmbH & Co. KG, Heidelberg, Ausgabe 3/2001, S.36–41
- Zipf, A., Chandrasekhara, V., Häußler, J. und Malaka, R.** *Deep Map – Mobiles GIS hilft Touristen bei der Navigation*; in: HGG-Journal (Journal der Heidelberger Geographischen Gesellschaft), Heft 14, Heidelberg, 2000, S.285–296