

---

# Low-Cost Modell zur Integration von Geodatenbeständen in objektorientierte Fachinformationssysteme

Thomas GUGGENBERGER

## Zusammenfassung

Die Zusammenführung komplexer naturwissenschaftlicher oder wirtschaftlicher Informationssysteme mit Geographischen Informationssystemen (GIS) ist nicht ohne weiteres möglich. Mit dem steigenden Bedürfnis nach räumlicher Interpretation von Fachdatenbeständen steigt aber die Notwendigkeit einer leistungsfähigen und funktionalen Verknüpfung vorhandener Systeme. Der vorliegende Beitrag beschreibt ein praktisches Architekturkonzept zur Überwindung der Grenzen zwischen GI-Systemen und Fachinformationssystemen (FIS). Die in einem Minimalfall benötigten Komponenten stammen aus der Open-Source-Welt und erzeugen somit keine zusätzlichen Kosten. Die Einführung eines speziellen Entwurfsmusters unterstützt die Umsetzung dieser Architektur.

## 1 Einleitung

Analysen von Datenstrukturen weisen fast immer auf einen Anteil räumlicher Informationen hin. Die Höhe des ermittelten Anteils und damit die Bedeutung von GIS hängen vom fachlichen Hintergrund des Analytikers ab. Geoinformatiker fokussieren stärker den räumlichen Zusammenhang, Datenbank-Administratoren bevorzugen Beziehungsmodelle (ER-Modelle). Beide müssen sich mit Datentypen und Organisationsstrukturen befassen und kommen im logischen Entwicklungskonzept häufig zum selben Ergebnis, obwohl ein unterschiedliches semantisches Verständnis vorliegt. Am Beispiel eines Punktes kann festgestellt werden: Geoinformatiker wissen, dass ein Punkt über seine Koordinaten gespeichert wird, ihnen ist aber die Funktion des Punktes (Teil einer Linie, ...) und dessen Beziehung (Topologie, ...) viel wichtiger. DB-Administratoren werden sehen eher das Tripel seiner Zahlenwerte, Entwickler allgemeiner Systeme interessieren sich für zusätzliche Information wie zum Beispiel eine Postadresse.

Die unter diesen Aspekten beschrittenen Entwicklungswege führten verfügbare Anwendungen zum Teil in die Isolation. So haben die Entwickler von Datenbanksystemen lange Zeit nicht an die Einführung eines Datentyps SPATIAL gedacht. (Wozu auch?!). Seitens der GI-Anwendungen kann bemerkt werden, dass wertvolle Entwicklungswerkzeuge des Software-Engineering erst langsam in die Programmumgebungen eingeführt werden.

Die Fähigkeiten von Software orientieren sich also offensichtlich am Bedarf der Anwendung. Gut so, denn überladene Programme verlieren ihre Leistungsfähigkeit. Für Geoinformatiker heißt dies aber, dass sie in der Lage sein müssen, Brücken zwischen unterschiedlichen Systemen zu bauen.

## 2 Kernproblem

Die Entwicklung von Schnittstellen ist ganz entscheidend an die Möglichkeiten der zu verknüpfenden Systeme gebunden. Ist seitens der GI-Systeme eine Veränderung des Programmcodes nicht möglich (trifft mit Ausnahme von GRASS-GIS immer zu), bleibt meist nur die Verwendung vorhandener Schnittstellenmethoden oder ein Zugriff auf die GI-Daten. Halten GI-Systeme die Daten in propriäteren Formaten ist nicht einmal dies möglich. Das gleiche gilt für FI-Systeme deren Quellcode aber deutlich häufiger verfügbar ist. Erst mit kreativen Eingriffsmöglichkeiten in das Fachdatenmodell werden die Überlegungen dieses Beitrags möglich! In der Planungsphase eines Fachmodells mit Geodatenchnittstelle sollten folgende fünf Fragen unter Berücksichtigung der zur Verfügung stehenden Systeme geklärt werden:

- Welche Möglichkeiten zur Koppelung bieten bestehende Systeme?
- Welche Architektur ist notwendig?
- Wird auf die Daten in uni- oder bidirektionale Richtung zugegriffen?
- Sollen auch Methoden der Partnersysteme genutzt werden?
- Wie wichtig ist das Verhalten hinsichtlich der Datenmenge?

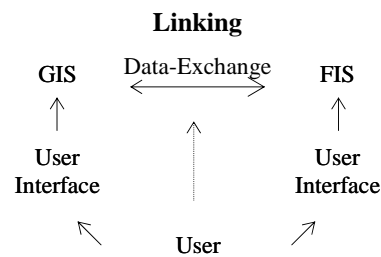
Sowohl die Problemanalyse als auch das entwickelte Konzept beruhen auf einem landwirtschaftlichen Forschungsprojekt, welches das gesamte Bundesgebiet von Österreich mit einer enormen Fach- (22 GB) und Geodatenmenge (70 GB) abdeckt. Erwartet werde Erkenntnisse über die Bewirtschaftung der Flächen (Ertrag, Nährstoffgehalt) und über die Haltung und Fütterung von landwirtschaftlichen Nutztieren (Düngeranfall, regionaler Verbrauch von Handelsfuttermittel).

Kernproblem des erwähnten Projektes war das nicht ausreichende Laufzeitverhalten des Fachdatenmodells innerhalb einer verfügbaren Programmierumgebung sowie die hohen Kosten für diese (ArcObjects).

## 3 Arten möglicher Datenverbindung

HARTKAMP et al. (1990) definiert für die Verbindung von GIS und FIS die Koppelung von Systemen über *linking* (Datenaustausch getrennter Systeme bei individueller Benutzerschnittstelle), *combining* (Datenaustausch getrennter Systeme mit gemeinsamer Benutzerschnittstelle) und *integrating* (Einbettung FIS in GIS oder umgekehrt). Die drei Varianten gehen davon aus, dass sich alle Datenquellen auf einem Rechner befinden. Ist das nicht der Fall, müssen *verteilte Anwendungen* entwickelt werden (KOKKONEN et al. 2003). Für die Anwendung der einzelnen Varianten wird in Abbildung 1 eine Bewertung und ein Beispiel gegeben. Muss die Systemarchitektur nicht verteilt ausgeführt werden, ist sicher das System der Integration von GIS und FIS innerhalb eines Werkzeuges der für den Entwickler idealste Fall. Limitierend wirkt hier aber wie als Kernproblem bereits erwähnt die Größe des Projektes. Für die landwirtschaftlichen Fachdaten wurde in einem relationalen Prototyp eine linear steigende Laufzeitkomplexität gemessen ( $O(t_n)=0.0001*n$ ). Die Ladezeit des ersten Objektes liegt bei 0, die des letzten bei 5 Minuten. Die Gründe dafür liegen einerseits in der Programmiersprache des Prototyps, vor

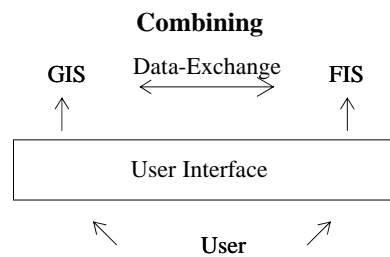
allein aber in den hohen Kosten der Datenabfragen innerhalb der relationalen Ausgangsdatenbanken des FIS.



**Vorteil:** Jedes System behält seine Eigenschaften und Leistung

**Nachteil:** Austausch kosten Zeit, nicht systematisierbar

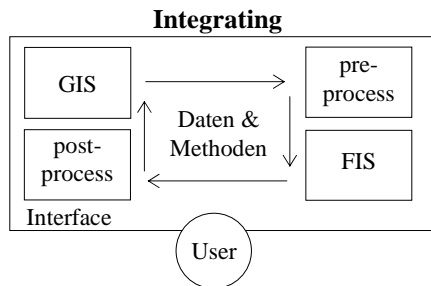
**Beispiel:** Join auf Daten via ODBC



**Vorteil:** Ein Interface garantiert systematische Abläufe

**Nachteil:** Austausch kosten Zeit, Methodenverlust im Interface

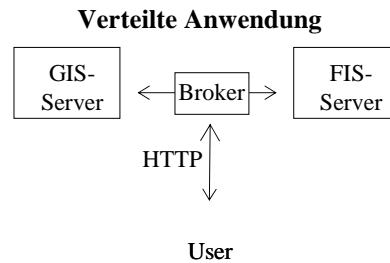
**Beispiel:** Individuelle Entwicklung mit z.B.: ArcObject



**Vorteil:** Direkter Datenzugriff, alle Methoden verfügbar

**Nachteil:** Großen Projekten nicht gewachsen (Leistung)

**Beispiel:** VBA-FIS in ArcGis



**Vorteil:** Ideale Interoperabilität, keine lokale Software

**Nachteil:** Komplexe Systeme die enorm teuer sind

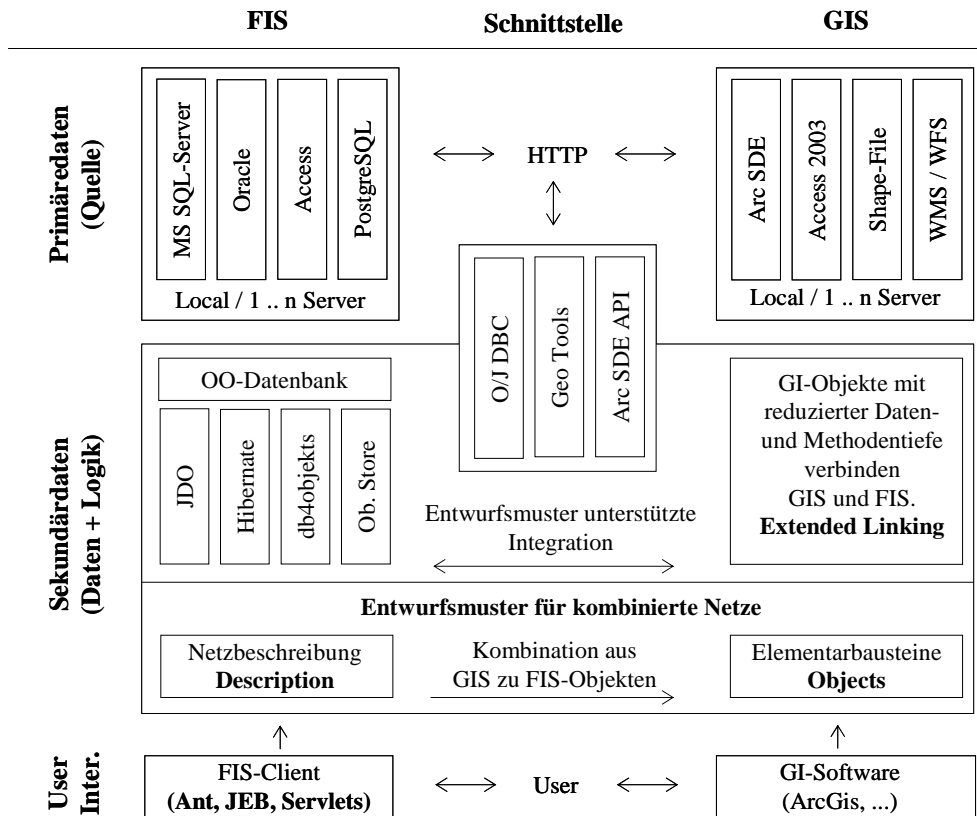
**Beispiel:** Manche Umweltinformationssysteme

**Abb. 1:** Klassische Varianten der Verbindung von Fach- und Geoinformationssystem (HARTKAMP et al. 1990, KOKKONEN et al. 2003, Huber 2005)

### 3 Architektur und Kosten des Fachinformationssystems

Der entscheidende Durchbruch innerhalb des Projektes konnte durch eine Kombination des Linking-Konzeptes mit der Architektur der verteilten Anwendung erreicht werden. Dabei wird innerhalb des FIS das relationale System der Ausgangsdaten durch ein objektorientiertes Konzept ersetzt. Entstehende Klassen werden als dauerhaft gültige

(persistente) Objekte in einer OO-Datenbank abgelegt. So erreicht die Laufzeitkomplexität einen konstanten Wert von 0.00045 Sekunden/Objekt. Der im Combining-Konzept noch als Data-Exchange dargestellte gegenseitige Austausch von Daten wird auch auf die Anwendung von Methoden des GIS (wenn vom GIS unterstützt), über komplexe Schnittstellenbibliotheken erweitert. Entscheidend ist dabei, dass diese Architektur in einem Minimalfall an Datenquellen ohne zusätzliche Softwarekosten erstellt werden kann (Kombination aus PostgreSQL, Geo-Tools mit der Programmiersprache Java unter Verwendung von Personal-Geodatabase oder Shape-Files als Geodatenquelle).



**Abb. 2:** Architekturschema zur OO-Vernetzung verteilter Datenquellen aus GIS und FIS.

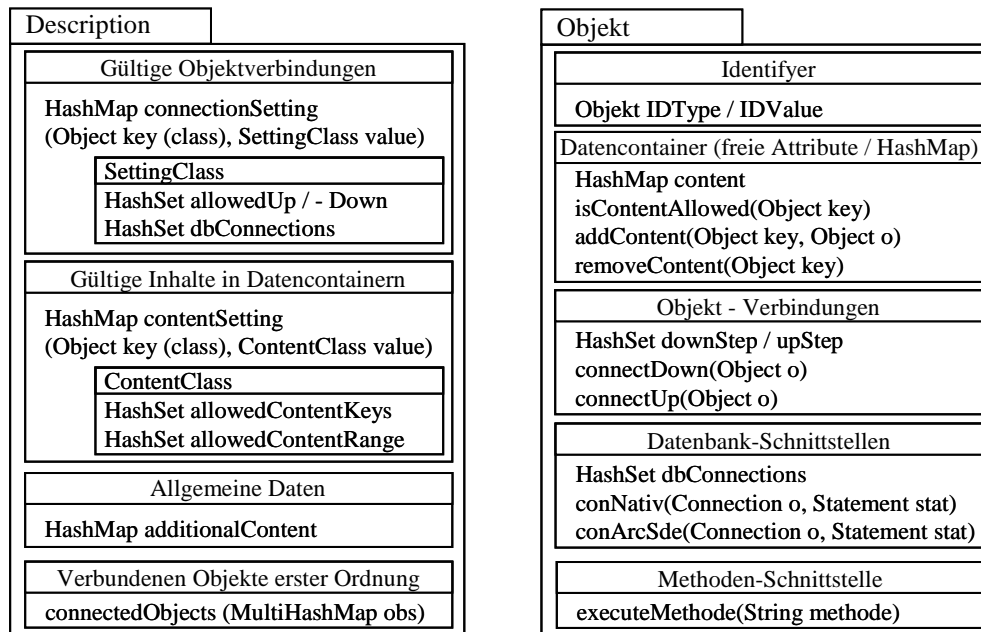
Die zentrale Aufgabe in Abbildung 2 liegt in der Entwicklung einer kombinierten Anwendung für GIS und FIS (Business-Logik in der Sekundärdatenschicht). Diese integriert über Schnittstellen die unterschiedlichsten Datentypen. Die Art und damit die Möglichkeiten der zu wählenden Softwareschnittstelle hängen von der jeweiligen Primärdatenquelle ab. Fachdaten aus relationalen Datenbanken werden via O/JDBC eingelesen, räumliche Daten je nach Primärquelle mit den Geo-Tools oder der ArcSDE-Schnittstelle. Entscheidend ist dabei, dass es innerhalb dieser Schicht zu einer netzartigen

Kombination aus Geo- und Fachdaten kommen kann. Die praktische Umsetzung kann mit dem ebenfalls entwickelten Entwurfsmuster für kombinierte Datennetze erfolgen.

## 4 Entwurfsmuster für kombinierte Datennetze

Auf der Grundlage einer elementaren Arbeit von COAD (1999) wurde diese Entwurfsmuster entwickelt und mit folgenden Eigenschaften ausgestattet:

- Durch die Einsatzmöglichkeit abstrakter Datentypen (ADT bzw. Klassen) kann das relationale Konzept überwunden werden. Eine vielschichtige Vermischung von fachlichen und räumlichen Objekten ist gut möglich (gleichzeitige Verwaltung von Bauernhof-Objekt und Grundstückspolygon), da diese vom selben Mutterobjekt abgeleitet werden.
- Hierarchische Strukturen bilden über direkte Speicherverweise die Zusammenhänge zwischen den Objekten ab. Relationale Konzepte wie Assoziationen aber auch räumliche Termini, wie Topologien können sich vermischen (topo-relationale Verbindung eines Bauernhof-Objektes mit dessen Grundstückspolygonen).
- Standardisierte Vererbung neuer Objektklassen von einigen wenigen Ausgangsklassen reduziert den Programmieraufwand auf ein Minimum.
- Beliebig erweiterbarer Zusammenhang zwischen hierarchischen Objektstrukturen (*Objektkette*) wie der Kette „Bauernhof – Grundstücke – Vegetation – Nährstoffkonzentration“ und zusätzlichen ebenso hierarchischen räumlichen Versorgungsstrukturen (*Versorgungsketten*) wie der Kette Bund – Hauptproduktionsgebiete – Kleinproduktionsgebiete – Gemeinden, bilden das gewünschte Informationsnetz.



**Abb. 2:** Aufbau der beiden Basisklassen *Description* und *Objekt* aus dem Paket *Objektkette* des Entwurfsmusters.

Die vier beschriebenen Eigenschaften werden in zwei Klassen abgebildet. Die beschreibende Klasse trägt die Bezeichnung *Description*. Sie definiert sowohl die innerhalb der Verknüpfung gültigen Objektverweise als auch die möglichen Attribute des Netzes. Die zweite Klasse mit der Bezeichnung *Objekt* erfasst diese und implementiert zusätzlich alle Schnittstellenmethoden. Ausgehend von diesen Klassen verbleibt für den interessierten Entwickler die Implementierung der speziellen Fachmethoden in den von *Objekt* abgeleiteten Klassen.

## 5 Umsetzung des Konzeptes

Sowohl die Architektur als auch das Entwurfsmuster wurden im Rahmen des erwähnten Projektes zum Teil umgesetzt und leisten gute Dienste (GUGGENBERGER 2005). Erste Ergebnisse liegen bereits vor und dienen derzeit dem Monitoring gesetzlicher Grundlagen wie der Nitratrichtlinie.

## 6 Literatur und Entwicklungswerkzeuge

ArcSDE API: Schnittstelle zur Geo-Database von ESRI für die Programmiersprachen Java und C. [http://edndoc.esri.com/arcscde/9.1/java\\_api/docs/overview-summary.html](http://edndoc.esri.com/arcscde/9.1/java_api/docs/overview-summary.html)

- Coad, P (1996): Modelling in Colour with UML. Prentic Hall, New York.
- Db4objects: OO-Datenbank. [www.db4objects.org](http://www.db4objects.org)
- Eclipse: Programmierschnittstelle für Java und Jdo. <http://www.eclipse.org>
- Geo-Tools: Schnittstellensoftware für GIS. <http://www.geotools.org/>
- Guggenberger, T, N. Bartelme (2005): GIS gestützte Modellierung der Nährstoffbilanzen Österreichischer Grünlandbetriebe. HBLFA Raumberg-Gumpenstein, Heft 43.
- Hartkamp et al (1999): Interfacing geographic information system with agronomic modeling: A review. In: Agronomy Journal, 91, S. 761-772.
- Huber, S, (2005): Numerische Modellierung und Simulation der räumlichen und zeitlichen Variabilität von Lachgasemissionen aus Agrarökosystemen. Dissertation, Universität Hohenheim.
- Java : OO-Programiersprache. <http://www.java.com>
- Jdo: Java Data Object. <http://jcp.org/en/jsr/detail?id=243>
- Kokkonen et al (2003): Interfacing environmental simulation models and database using xml. In: Environmental Modelling & Software, 18, S. 463-471.
- PostgreSql: Relationale Datenbank. <http://www.postgresql.org/>