

Integration dynamischer Atmosphärenmodelle mit einem (3+1)-dimensionalen objektorientierten GIS-Kern*

Ludger Becker¹, Lars Bernard², Jürgen Döllner¹,
Stefan Hammelbeck¹, Klaus H. Hinrichs¹, Thomas Krüger²,
Benno Schmidt² und Ulrich Streit²

Abstract

Efforts to integrate Geographic Information Systems (GIS) and environmental simulation models often fail due to the structure of current GIS which do not meet the requirements of complex numerical simulation techniques. Therefore research in the field of environmental modeling focuses on adequate flexible simulation methods and powerful interoperable GIS interfaces to implement these methods.

This paper shows approaches towards an interoperable object-oriented GIS interface for atmospheric modeling and analysis. The architecture and specifics of a first prototype system, called AtmoGIS, are presented.

1 Einleitung

Die Modellierung von Umweltprozessen erfordert häufig den Einsatz von Geoinformationssystemen (GIS) und dynamischen Umweltsimulationssystemen (Burrough et al. 1988). Die Kombination beider Systeme ist sinnvoll, da

- *Geoinformationssysteme*, insbesondere in Hinblick auf die fehlenden numerischen Fähigkeiten und die Modellierung der Zeit bzw. der dritten räumlichen Dimension, in der Regel nicht den Anforderungen komplexer Umweltsimulationsmodelle genügen und

* Dieses Projekt wird von der Deutschen Forschungsgemeinschaft unter der Projekt-Nr. STR 172/8-1 gefördert.

¹ Westfälische Wilhelms-Universität Münster, Institut für Informatik, Einsteinstraße 62, D-48149 Münster, email: {beckelu, dollner, hamstef, khh}@math.uni-muenster.de, Internet: <http://wwwmath.uni-muenster.de>

² Westfälische Wilhelms-Universität Münster, Institut für Geoinformatik, Robert Koch-Straße 26-28, D-48149 Münster, email: {bernard, benno, krugert, Streit}@ifgi.uni-muenster.de, Internet: <http://ifgi.uni-muenster.de>

- Simulationswerkzeuge wegen fehlender Verwaltungs- und Analysefunktionalitäten meist keinen angemessenen Umgang mit raumbezogenen Umweltdaten erlauben.

Die Verbindung von GIS und dynamischen Umweltsimulationsmodellen beschränkt sich derzeit meist auf die Entwicklung einer Schnittstelle zwischen zwei mehr oder weniger eigenständigen Softwaresystemen. Diese Anwendungen besitzen dabei jeweils eigene Datenmodelle, wodurch eine Datenkonvertierung zwischen den Systemen erforderlich wird. Diese Strategie wird nach Nyerges (1992) als *Kopplung* bezeichnet und ist gekennzeichnet durch redundante Datenhaltung, Semantikverlust und geringe Effizienz.

Zur Vermeidung dieser Nachteile konzentrieren sich aktuelle Forschungstätigkeiten auf die Realisierung von Systemumgebungen, die eine *Integration* dynamischer geowissenschaftlicher Modelle mit GIS-Funktionalitäten ermöglichen (Becker et al. 1996, Bernard et al. 1998b, Burrough/McDonnel 1998, Johnston et al. 1996). Die Anwendungen arbeiten dabei auf demselben Datenbestand und interagieren unter gegenseitiger Nutzung der von den Anwendungen zur Verfügung gestellten Methoden. Aus diesem Ansatz ergeben sich für die Entwicklung von GIS-Umgebungen und Umweltsimulationsmodellen zwei Anforderungen:

1. Realisierung von räumlichen und zeitvarianten (3D-) Datenstrukturen in GIS, da die heute verfügbaren kommerziellen Systeme auf statischen 2D-Daten basieren.
2. Entwurf und Entwicklung von *interoperablen* Komponenten, die je nach Bedarf zu neuen Anwendungen kombiniert werden können, ohne neue „Monolithen“ zu schaffen.

Die Entwicklung effektiv und flexibel nutzbarer GIS-Umgebungen, die eine zeit- und kosteneffiziente Implementierung integrierter Anwendungen ermöglichen (Buehler/McKee 1996) wird insbesondere durch die OpenGIS-Initiative (OGC 1998) als internationalem Zusammenschluß von GIS-Anwendern und namhaften Soft- und Hardware-Herstellern vorangetrieben.

Die durch ein Bündelverfahren von der Deutschen Forschungsgemeinschaft geförderte Initiative „Interoperable Offene Geowissenschaftliche Informationssysteme“ (IOGIS 1997) setzt sich ebenfalls die Konzeption interoperabler Systeme zum Ziel. IOGIS stellt insbesondere die Integration hochdimensionaler geowissenschaftlicher Modelle mit GIS und die Verwendung objektorientierter Techniken zur GIS-Entwicklung in den Mittelpunkt. Die hier vorgestellten Arbeiten sind Bestandteil des IOGIS-Bündels.

In diesem Beitrag werden die Konzeption und die prototypische Entwicklung einer objektorientierten, interoperablen GIS-Umgebung zur Simulation und Analyse atmosphärischer Prozesse (AtmoGIS) vorgestellt.

2 Systemarchitektur von AtmoGIS

AtmoGIS ist eine C++-Klassenbibliothek, die eine integrierte Nutzung von interoperablen Komponenten zur Verwaltung und Analyse von raumbezogenen Daten, zur Modellierung von atmosphärischen Prozessen und zur Visualisierung von dreidimensionalen, dynamischen Datensätzen ermöglicht. Abbildung 1 zeigt den modularen Aufbau des Gesamtsystems und die Integration der Einzelkomponenten zu Anwendungen für die wissenschaftliche und planerische Praxis. AtmoGIS basiert auf einem interoperablen GIS-Kern (siehe Abschnitt 3.1 und 3.2) und kommuniziert mit verschiedenen, austauschbaren Implementierungen des GIS-Kerns (GOODAC, FSGDM) über eine abstrakte objektorientierte Schnittstelle für rasterbasierte Geodaten (VirGIS).

Die Anbindung von Simulationsmodellen, wie z.B. SKIMO oder PDCA (Pollutant Dispersion Cellular Automata) an AtmoGIS erfolgt über verschiedene C++-Klassen, die das Grundverhalten eines Simulationslaufes sowie numerische Lösungsschemata zur Verfügung stellen (siehe Abschnitt 3.3). Zur Visualisierung der zeitlich dynamischen, dreidimensionalen Modelldaten wird in AtmoGIS das Visualisierungssystem MAM/VRS verwendet, das auch über die abstrakte Schnittstelle mit dem GIS-Kern zusammenarbeitet (siehe Abschnitt 3.4). Der Benutzerdialog kann durch verschiedene GUI-Toolkits wie z.B. Motif, die Microsoft Foundation Classes oder das Tcl-basierte Tk (Ousterhout 1994) in Kombination mit der objektorientierten Tcl-Erweiterung [incr Tcl] gestaltet werden.

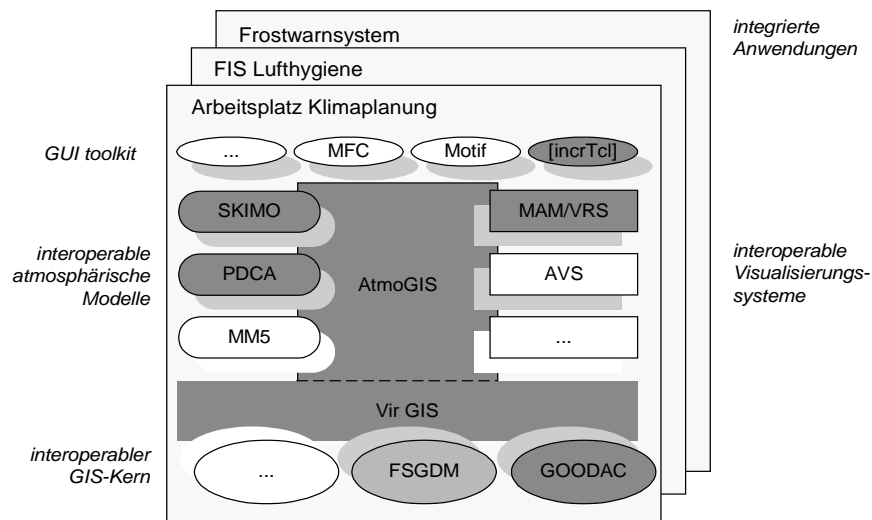


Abbildung 1

Systemarchitektur von AtmoGIS, realisierte und geplante Komponenten
(dunkel und hell unterlegt)

Die Interoperabilität zwischen den einzelnen Komponenten wird durch Anwendung objektorientierter Entwurfstechniken (Rumbaugh et al. 1991) erreicht. Die Schnittstellen der Systemkomponenten sind gekennzeichnet durch entsprechend modellierte, abstrakte Klassenhierarchien, deren Spezifikationen von integrierten, konkreten Implementierungen erfüllt werden. Anwendungen, die die Komponenten nutzen, arbeiten ausschließlich mit den Schnittstellenobjekten.

Im folgenden sollen die einzelnen Komponenten beschrieben und ihre Kommunikation im Bereich der Schnittstellen vorgestellt werden.

3 Die interoperablen Systemkomponenten von AtmoGIS

3.1 Die abstrakte VirGIS-Schnittstelle

VirGIS hat die Aufgabe, die raumbezogenen Objekttypen, die für Umweltsimulationsmodelle erforderlich sind, durch eine abstrakte Schnittstelle zu definieren. Dadurch wird gewährleistet, daß auf VirGIS basierende Fachanwendungen und GIS-Plattformen, die die VirGIS Schnittstelle anbieten, interoperabel sind. VirGIS stellt keine vollständige GIS-Schnittstelle zur Verfügung, sondern beschränkt sich auf die Bereitstellung einer leicht nutzbaren und erweiterbaren Schnittstelle zur Untersuchung der Interoperabilität.

Derzeit existieren eine dateibasierte und eine auf einem objektorientierten DBMS aufsetzende Implementierung der VirGIS-Schnittstelle. Die durch die VirGIS-Schnittstelle realisierte Abstraktion soll es dem Anwendungsentwickler ermöglichen, Applikationen unabhängig von der zugrundeliegenden Implementierung (Dateisystem oder Datenbank) zu entwickeln.

Im Zusammenhang mit der Integration von atmosphärischen Simulationsmodellen ist primär die Unterstützung von Gitterdatenstrukturen erforderlich. VirGIS umfaßt daher hauptsächlich Objekttypen für die Verwaltung und die Verarbeitung hochdimensionaler Gitter (Döllner/Bernard 1999). Eine Traversierung der Gitterdaten wird durch den Einsatz von Iteratoren ermöglicht. Der Datenzugriff wird über eine direkte Werteabfrage oder über verschiedene raum-zeitliche Interpolationsmethoden realisiert.

Mit den definierten raumbezogenen Objekttypen kann die VirGIS-Ebene als Basis für die Entwicklung GIS-gestützter Umweltsimulationsmodelle dienen. Darüber hinaus steht damit eine Testumgebung zur Verfügung, mit deren Hilfe Implementierungen von Spezifikationen aus dem Umfeld der objektorientierten GIS-Entwicklung (z.B. die Spezifikationen des OpenGIS-Consortiums; OGC 1998) erprobt werden können.

3.2 Implementierungen von VirGIS durch verschiedene GIS-Kerne

Eine der beiden aktuellen VirGIS-Implementierungen basiert auf dem objektorientierten Datenmodell OOGDM (Object-Oriented Geo Data Model, Abschnitt 3.2.1) und dessen prototypischer Implementierung GOODAC (Geo Object-Oriented Database Core, Abschnitt 3.2.1). Eine weitere VirGIS-Implementierung basiert auf FSGDM (File System Geo Data Management), einem Datenmodell, das primär zur dateibasierten Verwaltung 2-, 3- und 4-dimensionaler Gitter entworfen wurde und daher nicht die DBMS-Funktionalitäten von GOODAC zur Entwicklung von GIS-Anwendungen, wie z.B. Transaktionskonzepte, Integritätsbedingungen (Ditt et al. 1997) oder eine Abfragesprache, bietet. Die VirGIS-Klassen werden von OOGDM- bzw. FSGDM-Klassen abgeleitet. Im weiteren werden das Datenmodell OOGDM und seine Implementierung GOODAC näher vorgestellt.

3.2.1 Das Datenmodell OOGDM

OOGDM ist ein objektorientiertes Datenmodell, das als Basis für die Entwicklung von GIS-Anwendungen konzipiert wurde. Grundlage von OOGDM ist eine Klassenhierarchie, welche die in GIS-Anwendungen am häufigsten auftretenden Datentypen unterstützt. So umfaßt die Hierarchie Klassen für raster- und vektororientierte Daten im 2D- und 3D-Raum und zeitvariante Daten. Details des Datenmodells und Einzelheiten zur Integration der Zeitkomponente finden sich in (Voigtmann 1998) bzw. (Voigtmann et al. 1996). Grundsätzlich unterscheidet OOGDM zwischen zwei (orthogonalen) Zeitdimensionen: Die *Valid-time* beschreibt die Gültigkeit von Entitäten in der realen Welt, die *Transaction-time* die Gültigkeit von Entitäten in der Datenbank.

Die von GIS-Anwendungen häufig benötigten räumlichen Operationen sind in den OOGDM-Klassen als Methoden realisiert. Insbesondere werden

- geometrische Operationen (z.B. Berechnung der *Bounding Box*)
- topologische Prädikate und Operationen (*cross, overlap, touch, in, disjoint*) und
- Prädikate zur Ermittlung von Richtungsrelationen (z.B. Himmelsrichtungen)

angeboten (siehe Becker et al. 1996, Voigtmann et al. 1996 und Voigtmann 1998).

Für OOGDM ist eine Objektdefinitions- und Abfragesprache entwickelt worden. Die Definition von neuen Anwendungsklassen erfolgt in der Objektdefinitionssprache OOGDM-ODL in einem C++ ähnlichen Stil. Die SQL-ähnliche Abfragesprache OOGQL (Object-Oriented Geo Query Language) dient zur Abfrage der Datenbank und insbesondere zur einfachen Realisierung von Analysefunktionalitäten. OOGQL unterstützt die objektorientierten Konzepte von OOGDM sowie die räumlichen und zeitvarianten Datentypen. Beide Sprachen orientieren sich an den Konzepten der Objektdefinitions- und Abfragesprache, die im *Object Database Standard* (Catell

1996) definiert wurden. Die Abfrage zeitvarianter Daten ist analog zu TSQL-2 (Snodgrass 1995) realisiert.

3.2.2 GOODAC als prototypische Implementierung des Datenmodells OOGDM

Für das Datenmodell OOGDM liegt mit GOODAC (Becker et al. 1996, Voigtmann et al. 1997) eine prototypische Implementierung vor. Die Systemarchitektur von GOODAC besteht aus zwei Ebenen (siehe Abbildung 2), einer deskriptiven Ebene und einer Repräsentationsebene. Die deskriptive Ebene wird zur Entwicklung von GIS-Anwendungen verwendet. Während auf dieser Ebene OOGDM als Datenmodell verwendet wird, basiert die Repräsentationsebene auf dem Repräsentationsdatenmodell, das speicherungsnahe Strukturen, wie z.B. Indexstrukturen und Repräsentationen geometrischer Datentypen, beinhaltet.

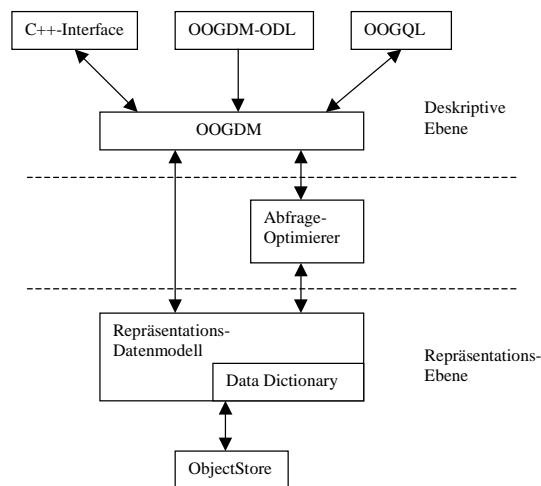


Abbildung 2
Die Systemarchitektur von GOODAC

Die Repräsentationsebene wurde auf dem objektorientierten Datenbanksystem *ObjectStore* (Lamb et al. 1991) implementiert, das die grundlegende Datenbankfunktionalität zur konsistenten Speicherung von Objekten zur Verfügung stellt.

Zur Entwicklung einer neuen GIS-Anwendung werden zunächst die (persistenten) Klassen in der OOGDM-ODL definiert und mit Hilfe eines Precompilers nach C++ übersetzt. Diese C++-Klassen stehen dann als Teil der deskriptiven Ebene für die Anwendungsentwicklung zur Verfügung. Der Anwendungsprogrammierer muß je-

doch noch den Code für die anwendungsspezifischen Methoden der neu definierten Klassen ergänzen sowie den allgemeinen Code für die Anwendung bereitstellen. Code zur Nutzung vordefinierter Funktionalität der OOGDM-Klassen und für den Zugriff auf die Attribute der neu definierten Klassen wird vom Precompiler automatisch bereitgestellt. Zusätzlich generiert der Precompiler Einträge für das Data Dictionary auf der Repräsentationsebene. Da VirGIS in seiner abstrakten Schnittstelle diese von GOODAC bereitgestellten Mechanismen nicht berücksichtigt, können diese in dem hier beschriebenen Kontext noch nicht eingesetzt werden.

GOODAC ist als offenes System konzipiert, um den Anforderungen möglichst vieler GIS-Anwendungen zu genügen. Neben der beschriebenen Erweiterbarkeit der Repräsentationsebene durch Ergänzung des Datenmodells um anwendungsspezifische Klassen und Funktionen kann der Prototyp auf der Repräsentationsebene um alternative Repräsentationen geometrischer Typen, Abfragemethoden, Indexstrukturen und Informationen für den Anfrageoptimierer erweitert werden. Die Erweiterungsmechanismen auf der Repräsentationsebene ermöglichen eine flexible Anpassung an die Bedürfnisse der unterschiedlichen Anwendungsklassen.

3.2.3 Integration und Interoperabilität

Da die AtmoGIS-Objekte auf VirGIS-Objekten basieren, die wiederum durch OOGDM- bzw. FSGDM-Objekte realisiert sind, arbeitet AtmoGIS unmittelbar auf den Datenstrukturen, die persistent gespeichert werden. Aufgrund der gewählten Systemarchitektur wird somit eine *Integration* zwischen AtmoGIS, VirGIS und den Implementierungen von VirGIS erzielt. Diese neuartige vollständige Integration von Anwendung und GIS vermeidet den bei der traditionellen Kopplung auftretenden Semantikverlust und gestattet den Verzicht auf modellspezifische Konvertierungsoperationen und Austauschformate.

VirGIS ermöglicht Interoperabilität in bezug auf das zugrundeliegende GIS. Auf VirGIS aufsetzende Anwendungen (wie z.B. AtmoGIS) benötigen kein spezielles Wissen über die benutzte Komponente, die die VirGIS-Schnittstelle implementiert.

3.3 AtmoGIS

AtmoGIS stellt eine Entwicklungsumgebung für atmosphärische Simulationsmodelle auf Basis der abstrakten VirGIS-Schnittstelle dar. Die Klassenbibliothek von AtmoGIS umfaßt

- Klassen zur Definition anwendungsspezifischer Objekte,
- Klassen zum Pre- und Postprocessing von Modelldaten und
- eine objektorientierte Simulationsumgebung für atmosphärische Modelle.

Die Integration atmosphärischer Simulationsmodelle erfolgt über ein Gerüst von Simulationsklassen, die das allgemeine Verhalten von Simulationsmodellen beschreiben.

Abbildung 3 gibt einen Überblick über die zentralen AtmoGIS-Klassen. Konkrete Implementierungen einzelner Simulationsmodelle werden durch die Spezialisierung der AtmoGIS-Klassen erreicht. Die Zusammenarbeit der einzelnen Klassen und der Ablauf einer Simulation sollen im folgenden kurz beschrieben werden.

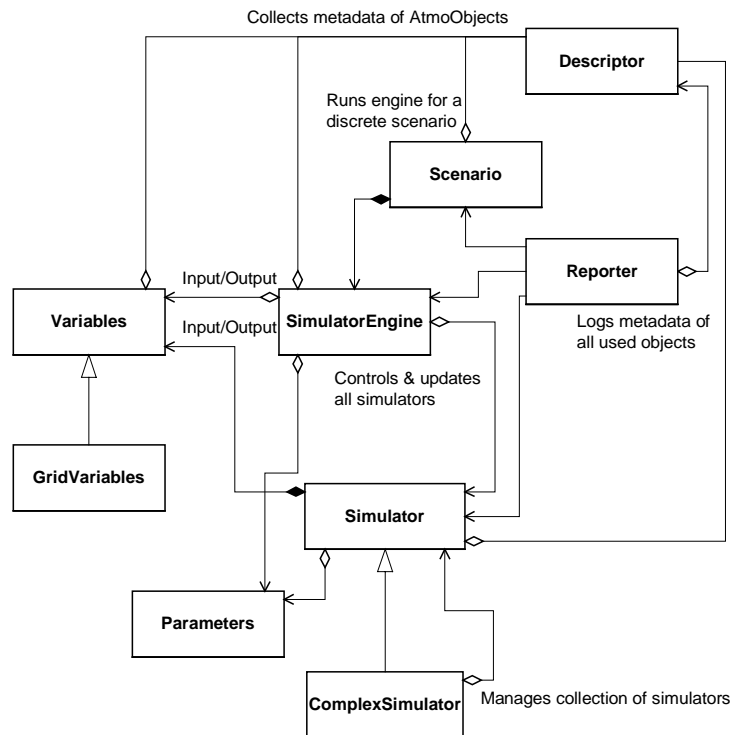


Abbildung 3
Übersicht über die elementaren AtmoGIS-Klassen

Zur Durchführung einer Simulationsrechnung wird eine Instanz von *Scenario* gebildet. Das *Scenario* übernimmt die Verwaltung und Initialisierung einer *SimulatorEngine*, die den zentralen Simulationszeitgeber darstellt und alle dem verwendeten Modell zugrundeliegenden Instanzen von *Simulator* verwaltet. Ein Modellauf wird mit dem Aufruf der *run()*-Methode der *Scenario* Klasse gestartet. Über die vorgegebene Simulationsdauer wird das Modell mit dem kleinstzulässigen Zeitschritt aller

beteiligten Simulatoren iteriert. In jedem Iterationsschritt erhalten alle Simulatoren von der *SimulatorEngine* eine *update()*-Nachricht. In Abhängigkeit von der aktuellen Simulationszeit können die Simulatoren selbst entscheiden, ob und wie eine Aktualisierung ihrer *Output*-Variablen möglich bzw. notwendig ist. Weiterhin bestimmen die Simulatoren den maximalen Zeitschritt, der in Übereinstimmung mit ihren Stabilitätsbedingungen für die nächste Zeititeration möglich ist. Diese Architektur ist damit auch geeignet, die Durchführung parallelisierter Simulationsanwendungen zu unterstützen. Der *ComplexSimulator* beinhaltet mehrere einzelne Simulatoren. So können beispielsweise die zur Lösung der Erhaltungsgleichungen notwendigen numerischen Schemata übersichtlich implementiert und verwaltet werden.

Ein Standardpool spezialisierter *Simulator*-Klassen für verschiedene Differenzverfahren (z.B. DonorCell, Upstream, etc.) und von Variablenklassen (z.B. Verwaltung von *staggered grids*, Verwendung orographiefolgender Koordinaten) sind im AtmoGIS bereits enthalten. Im Bedarfsfall kann dieser Pool durch die Anbindung einer leistungsstarken Numerik-Bibliothek, etwa des ebenfalls als C++-Implementierung vorliegenden DIFFPACK-Systems (NO 1998), erheblich vergrößert werden.

Dieses Simulatorkonzept wurde zur Integration eines mesoskaligen Atmosphärenmodells (SKIMO; Bernard 1995) eingesetzt. Darüber hinaus dient es zur Anbindung eines mikroskaligen Ausbreitungsmodells auf Basis Zellulärer Automaten (PDCA; Krüger 1999). Die jeweiligen Simulationsmodelle spezialisieren die in AtmoGIS zur Verfügung gestellten Simulationsklassen.

Der Zugriff auf die Modellvariablen erfolgt im gesamten Kontext des Modells einheitlich über die Klasse *Variables*, die als Input- oder Output-Objekt in den jeweiligen Simulatoren eingesetzt wird und die zeitlich dynamischen, dreidimensionalen Modelldaten organisiert. Diese Klasse verwaltet damit die Schnittstellenobjekte zum GIS-Kern. Spezielle Modellparameter können über die Klasse *Parameter* zur Verfügung gestellt werden.

Die in Abbildung 3 dargestellten *Descriptor*-Klassen verwalten die Metadaten der implementierten Simulationsbausteine. Jeder Simulator erzeugt bei seiner Instanziierung einen Deskriptor, der den Simulator textuell und formal beschreibt. Die in Textform erfolgten Beschreibungen dienen der späteren Szenario- und Modelldokumentation. Der *Reporter* iteriert über alle an der Szenariorechnung beteiligten Bausteine. Die textuelle Beschreibung kann dann als Modellreport ausgegeben werden.

Die Beschreibung der Simulatoren nach formalen Kriterien macht die Modellsemantik für Konsistenzkontrollen der instanziierten Simulationsmodelle und für Plausibilitätsüberprüfungen im Rahmen von Ergebnisanalysen verfügbar (Bernard/Pundt 1998). In dieser Form können beispielsweise die folgenden *constraints* im System formuliert und genutzt werden:

- Anforderungen einzelner Simulatoren an andere Simulatoren bzw. das gesamte Simulationsmodell ("*Simulator XY zur Druckberechnung setzt Annahme einer hydrostatischen Atmosphäre voraus*")
- in Verbindung mit den Modellannahmen minimal erlaubte raum-zeitliche Auflösungen für Ergebnisanalysen ("*Rasterweite bei Analyse muß > X m sein*")

Der Aufbau und die Systematisierung von entsprechenden Metadatenkatalogen für die Simulationsklassen können zukünftig eine geeignete und vollständige formale Beschreibung der Modellbausteine ermöglichen.

3.4 Das interoperable Visualisierungssystem MAM/VRS

Die Visualisierungskomponente ist ein zentraler Baustein einer GIS-Anwendung, da ihre Fähigkeiten über die Qualität der Benutzerschnittstelle entscheiden. Eine Reihe leistungsfähiger Techniken zur Visualisierung von Geodaten und ihrer Dynamik wurden entwickelt, deren Ziel es ist, Einsicht in die Geodaten zu ermöglichen und die Dynamik von Prozessen verstehbar werden zu lassen. Diese Visualisierungstechniken greifen intensiv auf Methoden und Techniken der 3D-Computergraphik zurück, die von einer Reihe von Visualisierungstoolkits bereitgestellt werden (z.B. Schroeder et al. 1996, Döllner/Hinrichs 1997).

Eine besondere Anforderung, die sich aus der Interoperabilität ergibt, besteht darin, ein Visualisierungssystem effizient mit den anderen interoperablen Systemkomponenten zu verbinden und das Visualisierungssystem den Bedürfnissen konkreter GIS-Anwendungen entsprechend zu erweitern. MAM/VRS, das in AtmoGIS als Visualisierungssystem verwendet wird, erfüllt die notwendigen Voraussetzungen zur effizienten Kopplung und gezielten Erweiterung.

MAM/VRS verfügt über Komponenten zur Modellierung von virtuellen Szenen auf der Basis von Szenengraphen, Komponenten zur Modellierung zeitlicher Abläufe auf der Basis von Verhaltensgraphen, wie sie bei der Computer-Animation benötigt werden, und Komponenten zur Spezifikation von Modell-Benutzer 3D-Interaktion. Zusätzlich können unterschiedliche 3D Rendering-Systeme, wie z.B. OpenGL oder RenderMan, zur Bildsynthese eingesetzt werden. Die Kriterien, die das Visualisierungssystem MAM/VRS für den interoperablen Einsatz qualifiziert, finden sich in all diesen MAM/VRS-Komponenten (Döllner/Hinrichs 1999) und sind im folgenden aufgeführt.

3.4.1 Nutzung von Applikationsdatenstrukturen für Rendering-Primitive

Die Graphikobjekte von MAM/VRS legen im allgemeinen nicht die interne Datenrepräsentation fest, sondern können auf applikationsspezifischen Datenstrukturen operieren. Technisch geschieht dies dadurch, daß Graphikobjekte auf der Basis von

Adaptieren bei Bedarf die notwendigen Daten aus der Applikationsdatenstruktur importieren. Die einfachste Form eines solchen Adapters ist der *Iterator*, ein Objekt, das den sequentiellen Zugriff auf eine Sequenz von Datenobjekten ermöglicht. Eine Applikationsdatenstruktur muß zum Einsatz in MAM/VRS eine Iteratorklasse definieren, die die Elemente einer solchen Datenstruktur zugreifbar macht. MAM/VRS Graphikobjekte benötigen keine Kenntnis, um welche Datenstruktur es sich handelt, da sie ausschließlich auf den Zugriffsmethoden der Adapterklassen arbeiten.

Als Beispiel sei das Mesh-Primitiv genannt, das durch ein zweidimensionales Feld von 3D-Punkten definiert wird. Das Mesh-Primitiv verlangt einen Iterator, der sequentiell, d.h. Reihe für Reihe und Spalte für Spalte, die 3D-Punkte bereitstellt. Dabei kann ein Iterator diese Punkte immer dann berechnen, wenn das Mesh-Primitiv diese benötigt, und dafür z.B. ein digitales Geländemodell aus einer Datenbank heranziehen oder die Daten prozedural berechnen.

Der Adapter-Ansatz trägt der Tatsache Rechnung, daß die GIS-Repräsentation in den seltensten Fällen als Graphik-Repräsentation herangezogen werden kann. Im Gegensatz zu einer Konvertierung, wie sie in traditionellen Ansätzen üblich ist, erlauben Adapter die funktionale und dynamische Verbindung beider Repräsentationsformen. Die Konvertierung und damit die redundante Speicherung in einem Zwischenformat entfallen. Außerdem bleibt die Semantik in der Visualisierungsrepräsentation erhalten, da mit Hilfe der Adapter die ursprünglichen Daten identifiziert werden können.

3.4.2 Erweiterbarkeit durch neue applikationsspezifische Graphik-Typen

Das Visualisierungstoolkit MAM/VRS definiert eine Reihe von Standard-Primitiven und graphischen Attributen in Form von Graphik-Klassen, die teilweise komplexe computergraphische Verfahren und Algorithmen kapseln. Die Ansteuerung wird durch diese Abstraktion im Vergleich zu einer direkten Programmierung mit einem 3D Rendering-System wesentlich vereinfacht. Da aber sowohl im Bereich der Computergraphik-Algorithmen als auch im Bereich der Visualisierungstechniken eine schnelle technische Entwicklung zu beobachten ist, stellt das Visualisierungssystem MAM/VRS explizit die Voraussetzungen bereit, um für einzelne Visualisierungsanwendungen spezialisierte Graphik-Primitive, Graphik-Attribute und Rendering-Algorithmen nahtlos in das Visualisierungssystem zu integrieren.

Für ein neues Graphik-Primitiv muß zum Beispiel durch Ableitung von einer geeigneten Basisklasse eine neue Graphik-Klasse erstellt werden. Zusätzlich können Zeichenalgorithmen und Schnittalgorithmen für den neuen Graphik-Typ definiert werden, die die spezielle Natur des Graphik-Typs ausnutzen. Darüber hinaus kann ein auf das zugrundeliegende 3D Renderingsystem zugeschnittener Rendering-Algorithmus definiert werden, um die optimale Rendering-Performance zu erzielen. Ein so neu integrierter Graphik-Typ unterscheidet sich daher nicht von den vordefi-

nieren. Hierin liegt ein wesentlicher Unterschied zu anderen Visualisierungssystemen, die nicht in diesem Maße konfigurierbar sind.

3.4.3 Integration von MAM/VRS mit AtmoGIS

AtmoGIS nutzt das Visualisierungssystem MAM/VRS zur Darstellung und Exploration von atmosphärischen Daten und ihrer Dynamik. Die Anbindung erfolgt durch spezialisierte Adapterklassen für die Gitterdatenstrukturen, auf denen AtmoGIS operiert.

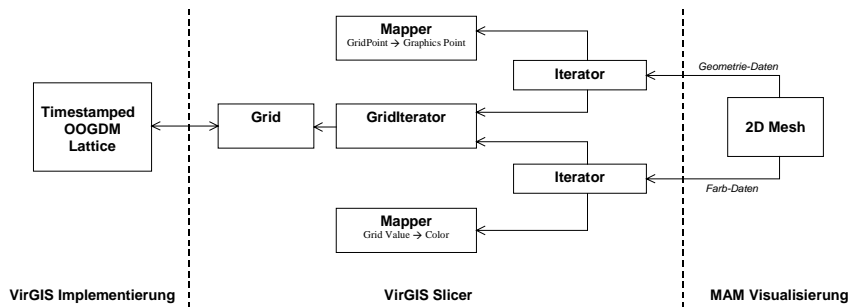


Abbildung 4
Implementierung eines Daten-Slicers mit Hilfe von MAM/VRS.

Als Beispiel für die Verbindung von Visualisierungssystem und AtmoGIS ist nachfolgend die Implementierung eines Daten-Slicers illustriert (siehe Abbildung 4). Bei den Daten handelt es sich um Temperaturwerte, die in einem dreidimensionalen Gitter angeordnet sind. Der Slicer wird repräsentiert durch ein zweidimensionales Mesh, das durch das dreidimensionale Gitter verläuft. Die Zellen des Slicer-Gitters beinhalten (durch VirGIS-Sampler berechnete) interpolierte Temperaturwerte, die zur Visualisierung in Farbwerte umgesetzt werden.

Zur Visualisierung werden zwei VirGIS-spezifische Iteratoren an ein 2D-Mesh angeschlossen. Die Iteratoren beziehen ihre Daten aus einem Grid-Iterator und bilden sie mit Hilfe von Mapper-Objekten auf 3D-Punkte bzw. Farbwerte ab. Der Grid-Iterator ist an ein zeitvariantes Gitter angeschlossen, das in diesem Beispiel eine OOGDM-Implementierung besitzt. Auf der Basis der VirGIS-Schnittstelle können Visualisierungstechniken und -strategien entwickelt werden, die sich unabhängig von der Implementierung der VirGIS Komponenten einsetzen lassen.

4 Zusammenfassung

Die hier beschriebenen Konzepte und Entwicklungen erlauben eine integrierte Nutzung dynamischer Umweltsimulationsmodelle und der zur Analyse der raum-zeitvarianten Simulationsergebnisse eingesetzten Werkzeuge. Insbesondere die durch die herkömmliche Kopplung von GIS und Modellen entstehenden Nachteile, nämlich redundante Datenhaltung, Semantikverlust und geringe Effizienz, werden dadurch verhindert.

Literaturverzeichnis

- Becker, L., Voigtmann, A., Hinrichs, K.H. (1996): Developing Applications with the Object-Oriented GIS-Kernel GOODAC, in: Proc. 7th Int. Symp. On Spatial Data Handling (SDH'96), Delft, The Netherlands, pp. 5A.1 – 5A.18
- Bernard, L. (1995): SKIMO-3 - Ein GIS-gestütztes dreidimensionales Mesoskalen-Modell zur numerischen Simulation der nächtlichen Wind-, Temperatur- und Feuchtefelder für den Einsatz in der Umweltplanung, in: Linnenberg W. (Hrsg.): Einsatz von DV-Methoden im Umweltbereich, Schriftenreihe des BDG, Heft 14, S. 95-97
- Bernard, L., Pundt, H. (1998): Semantikverlust in integrierten Systemen – Ein Fallbeispiel aus der Umweltplanung, in: Umweltinformatik 98, 12. Internationales Symposium "Informatik für den Umweltschutz", Bd. 2, Marburg, S. 528-540
- Bernard, L., Schmidt, B., Streit, U., Uhlenkücken, C. (1998b): Managing, Modeling, and Visualizing High-Dimensional Spatio-Temporal Data in an Integrated System, in: Geoinformatica, 2, No. 1, Norwell/MA, pp. 59-77
- Bernard, L., Schmidt, B., Streit, U. (1998a): AtmoGIS - Integration of Atmospheric Models and GIS, in: Proceedings of the 8th International Symposium on Spatial Data Handling (SDH '98), Vancouver, Canada, pp. 267-276
- Buehler, K., McKee, L. (1996): The OpenGIS Guide - Introduction to Interoperable Geoprocessing, Technical report, Open Geodata Interoperability Specification (OGIS), Open GIS Consortium, Inc.
- Burrough, P.A., McDonnel, R.A. (1998): Principles of Geographical Information System, Oxford
- Catell, R.G.G. (1996): The Object Database Standard ODMG-93, Release 1.2, San Francisco/CA
- Ditt, H., Becker, L., Hinrichs, K.H., Voigtmann, A. (1997): Constraints and Triggers in an Object-Oriented Geo Database Kernel, in: Proc. 8th Int. Workshop on Database and Expert Systems Applications (Workshop DEXA'97), Toulouse, France, pp. 508 – 515
- Döllner, J., Hinrichs, K.H. (1997): Object-oriented 3D Modeling, Animation, and Interaction, in: The Journal of Visualization and Computer Animation, 8, No. 1, pp. 33-64
- Döllner, J., Bernard, L. (1999): A Reference Model for N-Dimensional Grids, Bericht Nr. 11/99-I, Angewandte Mathematik und Informatik, Universität Münster

- Döllner, J., Hinrichs, K.H. (1999): An Object-Oriented Approach for Integrating 3D Visualization Systems and GIS, in: Computer & Geosciences, special issue "Geoscientific Visualization", in press
- IOGIS (1997): Interoperable Offene Geowissenschaftliche Informationssysteme (IOGIS), http://ifgi.uni-muenster.de/3_projekte/4dgis/texte/iogis/IOGIS.html
- Johnston, C.A., Cohen, Y., Pastor, J. (1996): Modeling of Spatially Static and Dynamic Ecological Process, in: Goodchild et al. (Hrsg.): GIS and Environmental Modelling, Fort Collins, pp. 149-154
- Krüger, T. (1999): Simulation der Ausbreitung von Schadstoffen in der Atmosphäre unter Einsatz zellulärer Automaten, IfGI prints 4, Institut für Geoinformatik, Münster (im Druck)
- Lamb, C., Landis, G., Orenstein, J., Weinreb, D. (1991): The ObjectStore Database System, in: Communications of the ACM, 34, No. 10, pp. 50 – 63
- NO (1998): Diffpacks User Guide, Numerical Objects
- Nyerges, T.L. (1992): Coupling GIS and Spatial Analytic Methods, in: Proc. 5th Int. Symposium on Spatial Data Handling, pp. 534-543.
- Schroeder, W.J., Martin, K.M., Lorensen, W.E. (1996): The Design and Implementation of an Object-Oriented Toolkit for 3D Graphics and Visualization, in: IEEE Proc. Visualization'96, pp. 93 – 100
- Snodgrass, R.T. (1995): The TSQL-2 Temporal Query Language, Dordrecht, The Netherlands
- OGC (1998): Open GIS Consortium, <http://www.opengis.org>
- Ousterhout, J. (1994): Tcl and the Tk Toolkit, Reading/MA
- Rumbaugh J., Blaha, M., Prmerlani, W., Eddy, F., Lorensen, W. (1991): Object-Oriented Modelling and Design, Englewood Cliffs/NJ
- Voigtmann, A., Becker, L., Hinrichs, K.H. (1996): Temporal Extensions for an Object-Oriented Geo-Data-Model, in: Proc. 7th Int. Symp. On Spatial Data Handling (SDH'96), Delft, The Netherlands, pp. 11A.25 – 11A.41
- Voigtmann, A., Becker, L., Hinrichs, K.H. (1997): Physical Design Aspects of Object-Oriented Geo Database Kernel, in: Proc. 8th Int. Workshop on Database and Expert Systems Applications (Workshop DEXA'97), Toulouse, France, pp. 529 – 534
- Voigtmann, A. (1998): An Object-Oriented Database Kernel for Spatio-Temporal Geo-Applications, Inaugural-Dissertation, Institut für Informatik der Westfälischen Wilhelms-Universität Münster