



**Forschungszentrum Karlsruhe**  
Technik und Umwelt

**Wissenschaftliche Berichte**  
FZKA 5623

**Integration einer Visualisierungs-  
komponente in ein  
wissenschaftliches Infor-  
mationssystem zur Beob-  
achtung atmosphärischer  
Phänomene**

**M. Wolber**

Institut für Angewandte Informatik

Oktober 1995

---



**Forschungszentrum Karlsruhe**  
**Technik und Umwelt**

**Wissenschaftliche Berichte**  
FZKA 5623

**Integration einer Visualisierungskomponente  
in ein wissenschaftliches Informationssystem  
zur Beobachtung atmosphärischer Phänomene**

Mechthild Wolber  
Institut für Angewandte Informatik

Forschungszentrum Karlsruhe GmbH, Karlsruhe

1995

Als Manuskript gedruckt  
Für diesen Bericht behalten wir uns alle Rechte vor

Forschungszentrum Karlsruhe GmbH  
Postfach 3640, 76021 Karlsruhe

ISSN 0947-8620

## Zusammenfassung

Dieser Bericht behandelt die Integration einer Visualisierungskomponente in ein wissenschaftliches Informationssystem zur Atmosphärenforschung auf der Basis eines Visualisierungs-Verwaltungs-Systems.

Die besonderen Anforderungen an diese Komponente sind einerseits ein informationsorientierter Datenzugriff mit einer fünfdimensionalen Zugriffsstruktur und andererseits die Zerteilung des Datenraums in explizit adressierbare, selbstbeschreibende Dateien, denen eine maximal dreidimensionale Datenstruktur zugrunde liegt.

Daher wird zunächst als Grundlage für die Eingaben an das Visualisierungssystem ein konzeptuelles Datenmodell entwickelt. Es folgt eine Vergleichsstudie bezüglich verschiedener Realisierungsszenarien für die räumliche Datenbank.

Anschließend wird eine hybride Lösung sowohl mit einem objektorientierten Datenbanksystem als auch mit einem wissenschaftlichen Datenformat unter Berücksichtigung der Einschränkungen durch die Schnittstelle zum Visualisierungssystem vorgestellt.

## Abstract

### **Integrating a Visualization Component in a Scientific Information System for Observation of Atmospheric Phenomena**

This report deals with the integration of a visualization component for a scientific information system, which is used within research analysing global change. This integration is based upon a visualization management system.

On the one hand, the special issue for this component is to provide an information-oriented data access using a five-dimensional data structure, whilst on the other, we need a division of the data space into explicitly addressable, self-describing files with a maximum of three dimensions. First, a conceptual data model is developed which is used to underly the input for the visualization system. Following on from this, different realization scenarios are then studied and compared for the spatial database.

Finally, a hybrid solution is presented using both an object oriented database system and a scientific data format, taking into consideration the restrictions of the interface to the visualization system.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Motivation . . . . .	7
1.2	Überblick . . . . .	9
<b>2</b>	<b>Problemfeld Informationssysteme - Visualisierungssysteme</b>	<b>11</b>
2.1	Das Gesamtsystem . . . . .	11
2.2	Die Visualisierungskomponente . . . . .	13
2.3	Integrationsaspekte . . . . .	16
2.4	Allgemeine Problematik . . . . .	18
<b>3</b>	<b>Das räumliche Datenmodell</b>	<b>20</b>
3.1	Warum fünf Dimensionen? . . . . .	20
3.2	Mehrdimensionale Datenstrukturen . . . . .	22
3.3	Konzeptuelles Modell . . . . .	23
<b>4</b>	<b>Realisierung durch ein Implementierungsmodell</b>	<b>26</b>
4.1	Relationale DBMS . . . . .	26
4.1.1	Datenmodellierung . . . . .	26
4.1.2	Zugriff auf die Daten . . . . .	27
4.1.3	Integrationsaspekte . . . . .	27
4.2	Objektorientierte DBMS . . . . .	28
4.2.1	Datenmodellierung . . . . .	28
4.2.2	Zugriff auf die Daten . . . . .	30
4.2.3	Integrationsaspekte . . . . .	31
4.3	Bewertung . . . . .	31
<b>5</b>	<b>Realisierung durch ein physikalisches Modell</b>	<b>33</b>
5.1	Wissenschaftliche Datenformate . . . . .	33
5.2	HDF . . . . .	35
5.2.1	Einführung in HDF . . . . .	35
5.2.2	Datenmodellierung . . . . .	36
5.2.3	Zugriff auf die Daten . . . . .	38
5.2.4	Integrationsaspekte . . . . .	38

5.3	NetCDF . . . . .	39
5.3.1	Einführung in netCDF . . . . .	39
5.3.2	Datenmodellierung . . . . .	39
5.3.3	Zugriff auf die Daten . . . . .	40
5.3.4	Integrationsaspekte . . . . .	40
5.4	Bewertung . . . . .	41
<b>6</b>	<b>Alternative: Hybride Lösung</b>	<b>43</b>
6.1	Konzept einer hybriden Lösung . . . . .	43
6.1.1	Architektur . . . . .	43
6.1.2	Datenhaltung . . . . .	44
6.1.3	Anfragebearbeitung . . . . .	46
6.2	Realisierung eines Prototyps . . . . .	50
6.2.1	Aufbau der räumlichen Datenbank . . . . .	50
6.2.2	Schnittstelle zum Visualisierungssystem . . . . .	53
6.3	Bewertung . . . . .	59
<b>7</b>	<b>Ausblick</b>	<b>61</b>
<b>A</b>	<b>Abkürzungsverzeichnis</b>	<b>63</b>

## Abbildungsverzeichnis

1	Das Gesamtsystem . . . . .	12
2	Föderative Architektur . . . . .	14
3	Die fünf Dimensionen . . . . .	20
4	Beispiele für Metadaten . . . . .	25
5	Beispielmodellierung für das relationale Modell . . . . .	26
6	Beispielmodellierung für das objektorientierte Modell . . . . .	29
7	Fünf-Schichten-Architektur für Datenbanksysteme . . . . .	34
8	Struktur einer Vgroup . . . . .	37
9	Beispielmodellierung mit HDF . . . . .	37
10	Beispielmodellierung mit netCDF . . . . .	40
11	Hybride Modellierung des VMS . . . . .	43
12	Architektur der räumlichen Datenbank . . . . .	45
13	Beispielmodellierung bei hybrider Lösung . . . . .	47
14	Anfragebearbeitung . . . . .	49
15	Objektklassen der räumlichen Datenbank . . . . .	50
16	Raumobjektstruktur . . . . .	52



# 1 Einleitung

## 1.1 Motivation

Die Lebensverhältnisse des Menschen sind seit Urzeiten von Wetter und Klima bestimmt. In steigendem Maße sind die natürlichen Schwankungen der meteorologischen Größen Strahlung, Temperatur, Bewölkung und Niederschlag von nachhaltigen anthropogenen Einwirkungen in die natürliche Umwelt überlagert, die durch Verkehr, Industrie und Landwirtschaft sowie insbesondere durch Nutzung fossiler Energieträger verursacht werden.

Besondere Gefahren ergeben sich aus dem stratosphärischen Ozonabbau, aus der Zunahme der Konzentration von klimawirksamen Spurengasen in der Atmosphäre und aus der großflächigen Versteppung, Verwüstung und Verkarstung von (Wald-)Gebieten.

Die Auswirkungen von anthropogener Erwärmung der Erde und Ozonabbau auf die natürlichen Ökosysteme sind komplex. Neben erhöhter UV-Strahlung sind ein Anstieg des Meeresspiegels und eine regionale Veränderung des Klimas (höhere Temperaturen, veränderte Niederschlagsmengen, mehr Stürme) zu erwarten.

Durch intensive Beobachtung der räumlichen und zeitlichen Verteilung meteorologischer Größen und der Spurengaskonzentrationen in der Atmosphäre sowie durch Laborversuche sind diejenigen Effekte untersucht worden, die für Treibhauseffekt und Ozonabbau verantwortlich sind.

Eindeutig nachgewiesen wurde der Abbau der stratosphärischen Ozonschicht durch FCKW-Emissionen. Dies ist besonders gravierend, da stratosphärisches Ozon durch seine Wirkung als Filter für die harte UV-Strahlung der Sonne Leben auf der Erde überhaupt erst möglich macht. Ebenso nachgewiesen wurde die ständige Zunahme der Konzentration klimawirksamer Spurengase ( $CO_2$ ,  $H_2O$ ,  $CH_4$ ,  $N_2O$ ,  $NO_x$ ,  $SO_2$ , ...) durch Verbrennung fossiler Brennstoffe, was einen Eingriff in den Wärmehaushalt der Erde über den atmosphärischen Strahlungshaushalt darstellt und eine globale Erwärmung befürchten läßt [Enq95].

Wissenslücken bestehen dagegen im Bereich der Wechselwirkungen der erwähnten Spurengase untereinander und mit der atmosphärischen Strahlung. Besonders wichtige Forschungsaufgaben sind die Erkundung von Kreisläufen relevanter Spurengase, der Einfluß von Wolken und Aerosolen auf den Strahlungshaushalt der Erde und die Dynamik des Stofftransports in der

Atmosphäre.

Das Verständnis der Wechselwirkungen und die Vorhersage der künftigen Klimaentwicklung wird durch ein leistungsfähiges Klimamodell ermöglicht [War91]. Zur Erstellung und Validierung eines solchen Modells sind neben strukturellem Wissen über atmosphärische Vorgänge enorme Mengen an Daten über den Zustand und die Zusammensetzung der Atmosphäre notwendig. Eine Möglichkeit zur Beschaffung solcher Daten ist die Fernerkundung der Atmosphäre durch Satelliten [FO90].

Für die von der ESA (European Space Agency) geplante Satellitenmission ENVISAT-1 wird deshalb am Forschungszentrum Karlsruhe im Rahmen des Vorhabens PROMISE im Institut für Angewandte Informatik in Zusammenarbeit mit dem Institut für Meteorologie und Klimaforschung ein wissenschaftliches Informationssystem zur Verwaltung, Auswertung und Analyse großer Mengen wissenschaftlicher Daten entwickelt.

Die Satelliten-Rohdaten stammen vom Instrument MIPAS (Michelson Interferometer for Passive Atmospheric Sounding). Dieses dient zur Messung von Stoffkonzentrationen einer großen Bandbreite relevanter Spurengase ( $O_3$ ,  $NO$ ,  $NO_2$ ,  $HNO_3$ ,  $N_2O$ ,  $N_2O_5$ ,  $CH_4$ ,  $ClONO_2$ ,  $F11$ ,  $F12$ ) sowie von Druck und Temperatur in der Stratosphäre und in wolkenfreien Regionen der oberen Troposphäre [Kra94]. Die Funktionsweise von MIPAS wird in [FO90] beschrieben.

Mit Hilfe des wissenschaftlichen Informationssystems sollen die MIPAS-Rohdaten zunächst zu höheren Datenprodukten (Interferogramme, Spurengasprofile, ...) verarbeitet werden. Dies geschieht im interaktiven Modus durch Experimentieren mit verschiedenen wissenschaftlichen Algorithmen und Validierung der erzeugten Datenprodukte [Kap94a].

Um ein besseres Verständnis der Vorgänge in der Atmosphäre zu erlangen, und um diejenigen menschlichen Einwirkungen identifizieren zu können, die für das künftige Klima die größten Risiken darstellen, sollen auf der Basis dieser Datenprodukte atmosphärische Phänomene beobachtet werden.

Unerlässlich für die Beobachtung und Analyse solcher Phänomene ist der Einsatz von Visualisierungstechniken, die es ermöglichen, große Datenmengen und komplizierte Sachzusammenhänge übersichtlich darzustellen ([EK94], [NB93]).

## 1.2 Überblick

Ziel dieser Arbeit ist die Entwicklung eines Konzeptes zur Integration einer Visualisierungskomponente in ein wissenschaftliches Informationssystem zur Atmosphärenforschung.

Die Gestaltung der Visualisierungskomponente hängt von den Anforderungen an das Gesamtsystem ab. Daher wird eine Einführung in die speziellen Anforderungen an ein wissenschaftliches Informationssystem zur Atmosphärenforschung und in das informationstechnische Konzept des Systems gegeben.

Für die Integration der Visualisierungskomponente in dieses System wird nach einer Bestandsaufnahme über verwandte Forschungsaktivitäten ein Kriterienkatalog erarbeitet, nach denen eine solche Integration vorgenommen werden kann (Kapitel 2).

Die Visualisierung atmosphärischer Phänomene erfordert einen informationsorientierten Zugriff auf die Datenmenge. Dadurch besteht die Möglichkeit, Datenbereiche deklarativ zu spezifizieren und an ein Visualisierungssystem zu übergeben.

Dies setzt aber die Existenz eines Datenmodells für die Verwaltung der zu visualisierenden Daten voraus. In der vorliegenden Arbeit wird ein fünfdimensionales Datenmodell für die Visualisierungskomponente entwickelt.

Neben diesem Modell werden die sich beim Entwurf des Visualisierungs-Verwaltungs-Systems auf der Grundlage dieses konzeptuellen Modelles aus der Mehrdimensionalität ergebenden Schwierigkeiten betrachtet (Kapitel 3).

Es folgt die Untersuchung und Bewertung der Realisierung des Visualisierungs-Verwaltungs-Systems durch relationale und objektorientierte Datenbanksysteme. Dabei wird einerseits auf die Datenmodellierung und auf effiziente Zugriffsmechanismen, andererseits auf die sich aus der jeweiligen Lösung ergebende Art der Integration fokussiert (Kapitel 4).

Als alternative Realisierungsmöglichkeit bieten sich wissenschaftliche Datenformate an. Die Realisierung des Visualisierungs-Verwaltungs-Systems durch die wissenschaftlichen Datenformate HDF und netCDF wird nach den oben genannten Kriterien untersucht und bewertet (Kapitel 5).

Das Ergebnis dieser Studie legt es nahe, einen Teil der räumlichen Datenbank mit einem wissenschaftlichen Datenformat zu realisieren und für den anderen Teil ein Datenbanksystem zu verwenden.

Diesem Ansatz folgend werden ein Konzept einer solchen hybriden Lösung sowie Strategien zu

Datenhaltung und Anfragebearbeitung entwickelt. Nach diesem hybriden Konzept wurde das Visualisierungs-Verwaltungs-System prototypisch implementiert (Kapitel 6).

Abschließend wird eine Zusammenfassung der Ergebnisse und ein Ausblick auf weitere zu lösende Probleme gegeben (Kapitel 7).

## 2 Problemfeld Informationssysteme - Visualisierungssysteme

### 2.1 Das Gesamtsystem

Wissenschaftliche Informationssysteme sind gekennzeichnet durch Mehrdimensionalität der Datenstrukturen und enorme Datenmengen. Herkömmliche Datenbanksysteme als alleinige Komponente sind nicht geeignet, solche Daten zu verwalten.

Ausgehend von den Anforderungen an ein solches System werden im folgenden die Architektur und die einzelnen Komponenten des wissenschaftlichen Informationssystems betrachtet.

Ziel der Beobachtung atmosphärischer Phänomene auf wissenschaftlicher Basis ist ein tieferes Verständnis der sich in der Atmosphäre abspielenden Vorgänge. Grundlage für das Erreichen dieses Verständnisses sind Meß- und Beobachtungsdaten, die über einen längeren Zeitraum hinweg in ausreichender Genauigkeit gesammelt werden.

Die wissenschaftliche Arbeit an sich besteht in der Erstellung von Taxonomien, experimentellen Gesetzen, Hypothesen und Theorien. Die Validierung und Erklärung von Hypothesen und Theorien führt zum Aufbau von Wissensstrukturen, die eventuell die Revision anderer Theorien herbeiführen ([Kap95], [KK95]).

Durch die Interdisziplinarität und Komplexität des Anwendungsgebietes bedingt, muß ein Informationssystem zur Atmosphärenforschung eine Plattform für die Arbeit von Wissenschaftlern verschiedener Fachgebiete zur Verfügung stellen. Hieraus ergeben sich die wichtigsten Anforderungen an das wissenschaftliche Informationssystem:

- Auswertung der Meßdaten (Interferogramme,...) zu höheren Datenprodukten (Spurengasprofile,...).
- Effiziente Verwaltung der generierten Daten. Dies ist besonders wichtig wegen der Größe der Datenmenge und der Vielfalt der vorkommenden Datenstrukturen.
- Erstellung von Hypothesen und Theorien, Analyse und Erklärung wissenschaftlicher Daten und Ergebnisse sowie wissenschaftliche Dokumentation.

- Erfassung der Generierungsgeschichte zur Rückkopplung auf die Ursprungsdaten für die Validierung beobachteter Phänomene.
- Visualisierung großer Datenmengen und komplizierter Sachzusammenhänge zur Gewinnung neuer Erkenntnisse.
- Das System sollte um Module zur multimedialen Erklärung und zur Simulation erweiterbar sein.

Zur Erfüllung dieser Anforderungen wurde in [Kap94b] ein wissenschaftliches Informationssystem entworfen, das den Prinzipien einer erweiterten föderativen Datenbankarchitektur genügt. Ein föderatives Datenbanksystem ist gekennzeichnet durch Heterogenität, Verteilung und Autonomie der Komponenten-Datenbanksysteme [LL94]. Ein erweitertes föderatives DBS kann außer aus Datenbanksystemen auch aus anderen Komponenten bestehen.

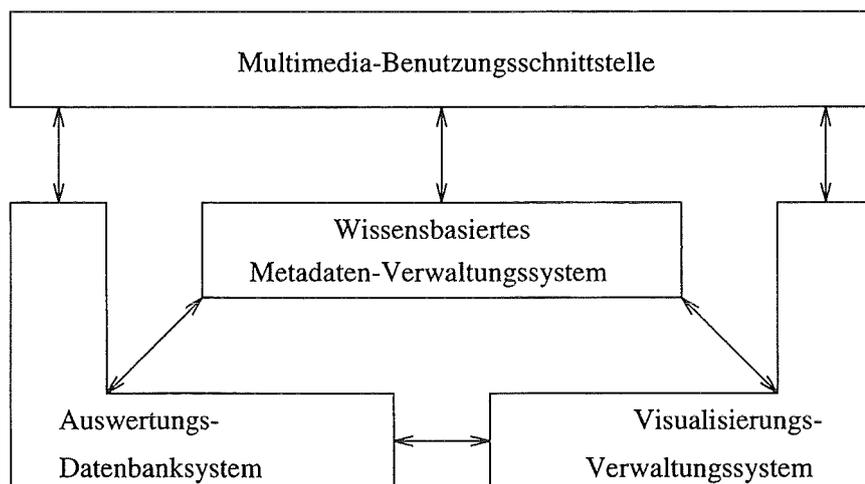


Abbildung 1: Das Gesamtsystem

Das wissenschaftliche Informationssystem besteht im wesentlichen aus

- wissensbasierter Metadatenverwaltungs-Komponente (Knowledge Based Management System - KBMS)
- Auswertungs-Komponente (Operational Database System - OpDBS und Mass Storage System - MSS)
- Visualisierungs-Komponente (Visualization Management System - VMS).

Die wissensbasierte Komponente dient zur Verwaltung und Generierung von Meta-Daten zur Analyse wissenschaftlicher Daten. Relevant ist hierbei sowohl Wissen, das sich auf die jeweilige wissenschaftliche Disziplin bezieht, als auch Wissen zur Modellierung des wissenschaftlichen Informationssystems und des speziellen Experiments.

Dieses Wissen ist nur teilweise a priori definiert. Ein anderer Teil muß aus den Komponenten OpDBS und VMS extrahiert werden. Die Realisierung der wissensbasierten Komponente erfolgt durch ein OODBMS. Das objektorientierte Datenmodell ist auch das globale Datenmodell des föderativen Datenbanksystems.

Die Auswertungskomponente unterstützt die Transformation und Generierung von höheren Datenprodukten aus Satelliten-Rohdaten. Eine spezielle Anforderung an ein wissenschaftliches Informationssystem ist die Revidierbarkeit und Regenerierbarkeit der Daten. Das bedeutet, daß zusätzlich zu den Meß- und Beobachtungsdaten auch deren Generierungshistorie erfaßt werden muß.

## 2.2 Die Visualisierungskomponente

In dieses Informationssystem zur Atmosphärenforschung soll nun eine Visualisierungskomponente integriert werden. Im folgenden werden die spezifischen Anforderungen für Entwurf und Realisierung dieser Komponente erarbeitet.

Die Visualisierungskomponente hat die Aufgabe, Phänomene aus den in Abschnitt 1.1 genannten Anwendungsbereichen zum Beispiel in Form von Zeitreihen zweidimensionaler Querschnitte aus dem dreidimensionalen Raum oder in Form von Volumenmodellen zu visualisieren.

Dazu sollen bereits existierende Visualisierungssysteme verwendet werden. Bei herkömmlichen Visualisierungssystemen erfolgt der Datenzugriff dateorientiert. Aufgrund der riesigen Datenmenge ist allerdings ein informationsorientierter Zugriff auf die zu visualisierenden Daten unerläßlich.

Benutzer müssen also die Möglichkeit haben, Datenbereiche durch deklarative Anfragen zu spezifizieren. Eine effiziente Abarbeitung solcher Anfragen setzt die Existenz eines Datenmodells mit fünf Dimensionen (atmosphärische Parameter ( $p$ ), Geokoordinaten ( $x,y$ ), Höhe ( $z$ ) und Zeit ( $t$ )) voraus.

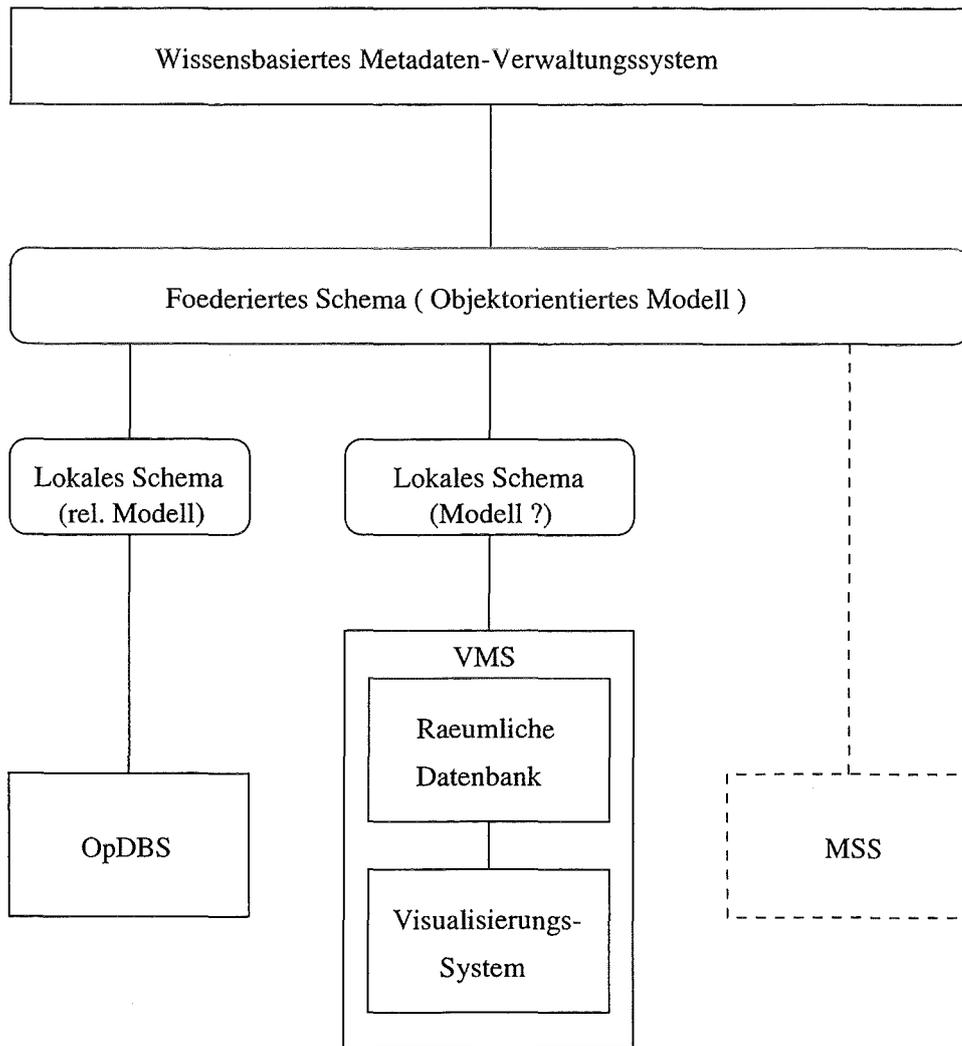


Abbildung 2: Föderative Architektur

Daher ist ein Visualisierungs-Verwaltungs-System zu entwerfen, das neben einem Visualisierungssystem aus der Instanziierung dieses fünfdimensionalen Modells, der räumlichen Datenbank (Spatial Database - SDB), besteht.

Die konkrete Aufgabe für die Gestaltung einer Visualisierungskomponente für das wissenschaftliche Informationssystem besteht in der Entwicklung dieser räumlichen Datenbank unter Berücksichtigung der Einschränkungen, die sich aus der Integration ergeben.

Die Integrationsanforderungen gliedern sich in zwei Teilbereiche: Einerseits ist die Schnittstelle zwischen Visualisierungs-Verwaltungs-System und Gesamtsystem zu betrachten, andererseits die Schnittstelle zwischen räumlicher Datenbank und Visualisierungssystem.

Die Integration des VMS ins Gesamtsystem muß nach den Prinzipien der Modularität, Flexibilität und Autonomie erfolgen. Das VMS muß sowohl in der Lage sein, autonom zu arbeiten, als auch ohne großen Aufwand in verschiedene Systemumgebungen eingebunden zu werden.

Wie die Integration erfolgen kann, hängt von der Abstraktionsebene des Datenmodells ab. In einem erweiterten föderativen Datenbanksystem sind dabei grundsätzlich zwei Möglichkeiten gegeben.

Ist die Komponente ein Datenbanksystem, so wird aus dem lokalen Schema ein Komponentenschema transformiert. Die lokalen Schemata werden dabei in eine einheitliche Darstellung überführt und um zusätzliche Informationen ergänzt. Hierfür wird ein Transformationsprozessor verwendet.

Wird die Komponente nicht mit einem Implementierungsmodell, sondern mit einem physikalischen Modell entworfen, so hat die Komponente kein lokales Schema. In diesem Fall ist lediglich ein Zugriffsprozessor notwendig.

An die Integration zwischen räumlicher Datenbank und Visualisierungssystem bestehen folgende Anforderungen:

- Benutzeranfragen spezifizieren meistens große Datenmengen. Deshalb ist ein effizienter Zugriff auf die Daten erforderlich.
- Es sollen verschiedene Visualisierungssysteme, wie z.B. Iris Explorer, AVS oder Vis5D, verwendet werden können. Diese unterstützen jedoch üblicherweise nur dreidimensionale Eingabedaten.
- Es müssen Heterogenität und Verteilung unterstützt werden. Verteilung ist für Visualisierungssysteme wegen des im allgemeinen riesigen Datenaufkommens sehr wichtig. Ebenso sinnvoll ist eine Verwendung von Spezialhardware für die Visualisierung. Deshalb ist auch

Heterogenität eine Eigenschaft, die unterstützt werden sollte. Beispielsweise könnte das Visualisierungssystem auf einer Plattform mit speziellen Graphikfähigkeiten installiert, die SDB aber auf einer anderen Plattform eingerichtet werden.

- Es soll ein Austausch von Daten mit anderen Wissenschaftlern möglich sein. Wünschenswert ist daher eine Speicherung der Daten in Form von selbstbeschreibenden Dateien.

### 2.3 Integrationsaspekte

Für den Entwurf der Integration eines Datenbanksystems mit einem Visualisierungssystem sind insbesondere folgende Aspekte zu betrachten [G<sup>+</sup>93]:

- Die Architektur der Integration.
- Der Grad der Datenkopplung.
- Das Datenmodell und die Anfragestrategie.

Die Architektur einer solchen Integration hängt stark von Verteilung und Heterogenität des Gesamtsystems ab. Grundsätzlich stehen zur Integration zwei Möglichkeiten zur Verfügung. Eine enge Hardwarekopplung ist sinnvoll, wenn explizit Single-User-Access gefordert ist, da dadurch gegenüber einer Client-Server-Architektur höhere Leistung und bessere Zuverlässigkeit erreicht werden kann.

Für die konkrete Anwendung bietet sich dagegen eine Client-Server-Architektur mit Visualisierungssystem als Client und Datenbank als Server an, da sie die Verteilung von Diensten, Multi-User-Access und, bei visualisierungsspezifischer Hardware, eine bessere Visualisierungsqualität erlaubt. Problematisch sind hierbei eventuelle Netz-Engpässe.

Die Datenkopplung zwischen Datenbanksystem und Visualisierungssystem kann eng oder lose sein. Dies wirkt sich sowohl auf den Zugriff des Benutzers auf die zu visualisierenden Datenbereiche als auch auf die Import/Export-Möglichkeiten aus.

Eine enge Datenkopplung erfordert hauptsächlich auf Serverseite Datenverwaltungsfunktionen. Alle Benutzeranfragen resultieren in einem Retrieval von der Datenbank, das DBMS regelt alle

Datenflüsse in der Applikation. Eine derartige Datenkopplung setzt allerdings eine Zerteilung der Datenmenge in Dateien, die direkt ans Visualisierungssystem geschickt werden können, voraus.

Bei loser Datenkopplung wird eine Untermenge der Datenbasis, die durch eine Anfrage spezifiziert wurde, in die Applikation des Benutzers geladen und dort bearbeitet. Dadurch werden sowohl auf Client- als auch auf Server-Seite Datenverwaltungsfunktionen wie Speicherung, Indizierung und Anfragebearbeitung benötigt.

Diese Lösung hat den Nachteil, daß die Eingabedaten an das Visualisierungssystem aus Kopien von Teilen des Datenbestandes bestehen. Da sich Benutzeranfragen zur Beobachtung atmosphärischer Phänomene normalerweise auf große Ausschnitte aus dem Datenraum beziehen, kommt eine lose Datenkopplung für das VMS aus Performanzgründen nicht in Frage [Koc93].

Datenmodelle dienen zur Abstraktion, Klassifizierung und Organisation von Daten. Beim Entwurf eines Datenmodells muß sowohl die Implementierung als auch die Nutzung des Visualisierungs-Verwaltungs-Systems berücksichtigt werden. Im Idealfall ist die Divergenz der sich aus beiden Anforderungssichten ergebenden Modelle minimal.

Ein Datenmodell muß eine funktionale Struktur sowohl zur logischen als auch zur physikalischen Organisation der Daten zur Verfügung stellen sowie ein Schema, das beide verbindet. Generell werden drei Abstraktionsebenen von Datenmodellen unterschieden [B<sup>+</sup>93]:

- Physikalische Datenmodelle: Zu ihnen zählen die wissenschaftlichen Datenformate. Bei herkömmlichen Datenbanksystemen ist das physikalische Datenmodell normalerweise unsichtbar.
- Implementierungsdatenmodelle: In dieser Arbeit werden das relationale und das objektorientierte Modell betrachtet.
- Konzeptuelle Datenmodelle: Diese dienen zur benutzernahen Beschreibung einer Datenbasis. Beispiel: Entity-Relationship-Modell.

Die Anfragestrategie leitet sich aus der Struktur der Daten ab. Diese wird durch die syntaktischen Metadaten beschrieben. Zusätzlich werden Metadaten zur Erhöhung der Leistung, z.B. durch Speicherung vorherrschender Zugriffspfade, und semantische Metadaten für die Bearbeitung intuitiver Anfragen benötigt.

Die Ausgestaltung der Integration von räumlicher Datenbank und Visualisierungssystem bezüglich Datenmodell und Anfragestrategie wird in den Kapiteln 4-6 im Hinblick auf die Anforderungen durch die konkrete Problemstellung weiter behandelt.

## 2.4 Allgemeine Problematik

Welche Funktionalität bieten existierende Systeme, die für die Realisierung eines Systems zur Visualisierung großer Mengen mehrdimensionaler Daten in Frage kommen?

Datenbanksysteme haben nur äußerst geringe Visualisierungsfähigkeiten. Datenbankapplikationen unterstützen allerhöchstens graphische Veranschaulichungen wie Balken- und Kuchendiagramme oder zweidimensionale Rastergraphiken.

Bestehende Visualisierungssysteme haben dagegen nicht die Fähigkeit, Daten zu verwalten. Sie können lediglich Daten aus einer Eingabe-Datei lesen.

Wissenschaftliche Datenformate (siehe Kapitel 5) bieten zwar die Möglichkeit der Übergabe von Dateien an ein Visualisierungssystem, erlauben jedoch außerhalb von spezifizierten Dateien keinen informationsorientierten Datenzugriff.

Diese Problematik hat zur Entwicklung eigener geographischer Informationssysteme (GIS) geführt. In den meisten existierenden GIS wird die reale Welt in eine zweidimensionale Betrachtung des Raums und in eine eindimensionale Betrachtung der zugehörigen Attribute aufgetrennt. Wenn zu den Dimensionen Länge und Breite die Höhe als Attribut gespeichert ist, spricht man auch von einem zweieinhalbdimensionalen Datenmodell.

Geographische Informationssysteme, die auf einem zweieinhalbdimensionalen Datenmodell beruhen, sind allerdings nur für Anwendungen z.B. aus der Raumplanung geeignet, die einen flächenbezogenen, sich nur langsam ändernden oder statischen Datenbestand verwalten ([Arm88], [BF91]).

Ein System, das sowohl eine für ein wissenschaftliches Informationssystem geeignete Datenverwaltung zur Verfügung stellt, als auch die Visualisierung von Volumendaten, Oberflächenmodellen und bewegten Bildern unterstützt, kann durch Integration eines Datenbanksystems mit einem Visualisierungssystem erstellt werden.

Die Forschungsaktivitäten bezüglich der Integration von Datenbanksystemen und Visualisierungssystemen sind aus zwei Gründen recht spärlich:

Erstens besteht ein Impedence Mismatch zwischen den für die Anwendung (Visualisierung) benötigten Daten und den in der Datenbank gespeicherten Daten: Wissenschaftliche Informationssysteme erfordern mehrdimensionale Datenstrukturen und Zugriffsmuster, Visualisierungssysteme dagegen akzeptieren nur Dateien mit maximal dreidimensionalen Daten als Eingabe.

Zweitens wird die Performanz von Visualisierungssystemen durch Datenbankverwaltungsoperationen erheblich verschlechtert.

Neben den Ansätzen, die Informationssysteme und Visualisierungssysteme lose koppeln und daher für das Visualisierungs-Verwaltungs-System von vornherein uninteressant sind (siehe Abschnitt 2.3), verfolgen bestehende Lösungsansätze zur Kopplung von Informationssystemen und Visualisierungssystemen zwei Strategien:

Ein Ansatz besteht in der Speicherung des Datenraums in Form von Dateien in einem wissenschaftlichen Datenformat und der Verwendung einer darauf aufgesetzten graphischen Metadaten-Verwaltungs-Oberfläche. Diese Lösung bietet allerdings keinen wirklich informationsorientierten Zugriff auf die Daten.

Der andere Lösungsansatz besteht in einer engen Kopplung zwischen einem Datenbanksystem und einem Visualisierungssystem. Da eine wichtige Anforderung an die Visualisierungskomponente die Unabhängigkeit von einem konkreten Visualisierungssystem ist, scheidet dieser Ansatz ebenfalls aus. Negativ zu bewerten sind außerdem die Möglichkeiten für den Austausch von Daten mit anderen Wissenschaftlern [Lee93].

Die besondere Schwierigkeit und Komplexität der Aufgabenstellung kommt also daher, daß einerseits die Forderung nach einem informationsorientierten Zugriff auf den Datenraum mit Hilfe fünfdimensionaler Zugriffsstrukturen besteht, und andererseits wegen der Möglichkeit zur Verwendung verschiedener Visualisierungssysteme und Hardwareplattformen sowie für Import/Export von Daten eine Zerteilung des Datenraums in explizit adressierbare, selbstbeschreibende Dateien mit maximal dreidimensionalen Datenstrukturen gefordert ist.

## 3 Das räumliche Datenmodell

### 3.1 Warum fünf Dimensionen?

Die Visualisierung der Ausbreitung und chemischen Interaktion von Gasen in der Atmosphäre erfordert die Möglichkeit zur räumlichen und zeitlichen Analyse großer Mengen an Meßdaten in unterschiedlicher Auflösung. Für die Beobachtung atmosphärischer Phänomene sind die einzelnen atmosphärischen Parameter in Abhängigkeit von Raum und Zeit zu betrachten.

Solche Problemstellungen können nur mit Hilfe eines Datenmodells mit den fünf Dimensionen atmosphärische Parameter ( $p$ ), Geokoordinaten ( $x,y$ ), Höhe ( $z$ ) und Zeit ( $t$ ) zufriedenstellend gelöst werden ([BK<sup>+</sup>91], [HLW91], [HMN93]).

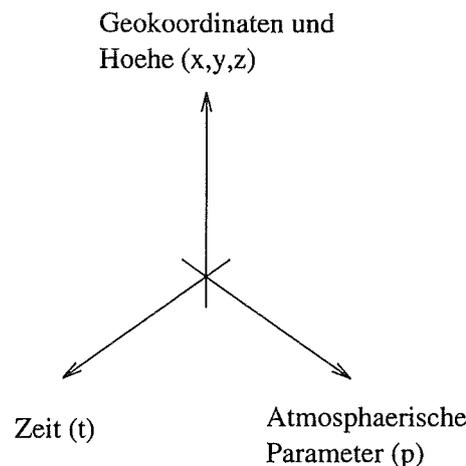


Abbildung 3: Die fünf Dimensionen

Eine theoretische Grundlage für dieses Modell bietet ein Gitter-Modell, das Extended Lattice Data Structure Model [KBS93]. Dies ist ein um syntaktische und semantische Metadaten ergänztes Datenstrukturmodell.

Ein Lattice (Gitter) mit den drei Komponenten Daten, Topologie und Geometrie beschreibt die Daten und deren Beziehungen untereinander. Dabei definiert die Topologie die Konnektivität zwischen Datenelementen, die Geometrie beschreibt die Projektion der Topologie in den

geometrischen Raum.

Wissenschaftliche Daten werden durch Beobachtung von Punkten in einem n-dimensionalen Raum gesammelt. In Anspielung auf diesen n-dimensionalen Raum wird das zu entwerfende fünfdimensionale Datenmodell im folgenden räumliches Datenmodell genannt.

Die räumliche Datenbank muß Funktionalitäten zur Verwaltung und Analyse dynamischer Volumendaten bieten, die nur auf der Grundlage eines solchen räumlichen Datenmodells gewährleistet werden können ([Bil93], [ATB90]):

- Alle fünf Dimensionen atmosphärische Parameter (p), Geokoordinaten (x,y), Höhe (z) und Zeit (t) müssen als Indizierungs-, Speicher- und Suchkriterium auf dem Datenbestand dienen. Dadurch läßt sich eine effiziente Anfragebearbeitung auch bei großen Datenmengen realisieren.
- Zur Analyse der Daten sollen räumliche und zeitliche Interpolationen, Ausbreitungsmodelle und Zeitsequenzen berechnet werden. Die Berechnung von Objektüberlagerungen ist vor allem bei Einbindung kartographischer Modelle für die Abarbeitung deklarativer Anfragen interessant.
- Mit Hilfe von Maßfunktionen sollten Nachbarschafts- und Konnektivitätsprobleme sowohl bezüglich der räumlichen Dimensionen als auch der Dimension Zeit gelöst werden können.
- Durch Animationen und Simulationen mit Hilfe einer Multimedia-Komponente muß die Visualisierung beliebiger Ausschnitte aus dem fünfdimensionalen Datenraum möglich sein. So können beispielsweise Zeitfolgen beliebiger zweidimensionaler Querschnitte entlang der drei räumlichen Dimensionen visualisiert werden.

Die Realisierung der räumlichen Datenbank mit Hilfe mehrdimensionaler Datenstrukturen ist für eine effiziente Verwaltung, Analyse und Visualisierung eines enormen Bestandes dynamischer Volumendaten unabdingbar.

Eine Abschätzung der Größe der zu verwaltenden Datenmenge ist jedoch selbst bei alleiniger Betrachtung der abzuspeichernden Meßdaten schwierig. Für die räumlichen Dimensionen ist folgende Grund-Granularität vorgesehen:

Länge x Breite x Höhe = 2.5 Grad x 2.5 Grad x 2.5 km. Dies ergibt bei Betrachtung von Teilen der Stratosphäre und Troposphäre (10 bis 50 km Höhe) über der gesamten Erdoberfläche insgesamt  $144 \times 72 \times 16 = 165\,888$  zu betrachtende Bereiche des Raums.

Unter der Annahme, daß über einen Zeitraum von fünf Jahren hinweg jeden Tag zu zehn atmosphärischen Parametern je ein Meßwert für alle diese Raumbereiche gesammelt wird, kommt

man bei einem Speicherplatzbedarf von 2 byte pro Meßwert zu einem Gesamtspeicherbedarf (nur für die Primärdaten!) von  $165\,888 \times (5 \times 365) \times 10 \times 2 \text{ byte} > 6 \text{ Gbyte}$ .

Im Vergleich zu herkömmlichen kommerziellen Anwendungen liegen die besonderen Anforderungen an das Visualisierungs-Verwaltungs-System insbesondere in der Beherrschung der Mehrdimensionalität und der enormen Datenmenge ([A<sup>+</sup>92], [AO93], [B<sup>+</sup>89]).

## 3.2 Mehrdimensionale Datenstrukturen

Für zweidimensionale Datenstrukturen gibt es einige Ansätze zur Lösung der Probleme bei der Manipulation der Datenstruktur. Zu diesen Problemen gehören der Aufbau einer solchen Struktur aus Eingabedaten, das Auffinden von interessierenden Teilbereichen aus der Datenbasis sowie die Überlagerung und Interpolation von Datenbereichen ([LT92], [Sam89]).

Eine Realisierungsmöglichkeit besteht in der Verwendung mehrdimensionaler Hash-Strukturen, sogenannter Grid-Files.

Eine Grid-File besteht aus einer Menge von Bereichen, die aus einer Zerteilung des Datenraums entlang der Dimensionen hervorgeht. Partitionierungen beziehen sich immer auf den gesamten Datenraum. Sie müssen nicht unbedingt äquidistant vorgenommen werden. Jeder Block zeigt auf ein Bucket, in dem die Datensätze gespeichert sind. Zur Verwaltung der Beziehungen zwischen Blocks und Buckets wird eine Katalogstruktur benötigt.

Grid-Files besitzen außerordentlich gute Zugriffseigenschaften. Allerdings ist die Balancierung eines Grid-Files im mehrdimensionalen Fall sehr kompliziert, das Verschmelzen von Blocks dagegen recht einfach.

Mehrdimensionale Suchbäume bieten die Möglichkeit eines hierarchischen Zugriffs auf die Daten, was zur Realisierung unterschiedlicher Auflösungen verwendet werden kann.

Für drei Dimensionen heißen diese Suchbäume Octrees. Allerdings ist es recht aufwendig, einen Octree aus einer Array-Darstellung zu konstruieren. Mit Octrees kann der Zugriff auf die Blocks nach drei Strategien erfolgen [Nie89]:

- Breitensuche: Jeder Knoten besitzt als Adresse eine ganze Zahl. Ausgehend von einem Knoten  $i$  werden als nächste die Knoten  $8i+1, \dots, 8i+8$  adressiert.

- Pfadadressierung: Innerhalb jedes Knotens wird der Raum entlang jeder Achse halbiert. Nach Knoten  $i$  werden also die Knoten  $i.1, \dots, i.8$  adressiert.
- bit interleaving: Die Zellen werden entlang jeder Achse anhand einer Binärzahl geordnet. Für drei Achsen mit jeweils vier Partitionen gilt: Wenn  $x = x_0x_1, y = y_0y_1, z = z_0z_1$  mit  $x_i, y_i, z_i$  aus  $\{0,1\}$ , dann ergeben sich als Zelladressen  $a = x_0y_0z_0x_1y_1z_1$  mit  $x_i, y_i, z_i$  aus  $\{0,1\}$ . Werden die so erhaltenen Adressen aufsteigend sortiert, so erhält man eine Peano-Kurve.

Eine Alternative zu Grid-Files und Octrees sind K-D-B-Bäume. Bei diesen werden nur überlaufende Seiten gesplittet. Da sie jedoch bedeutend schlechtere Zugriffseigenschaften haben, kommen sie für das Datenmodell des VMS nicht in Frage.

Prinzipiell lassen sich alle diese Ansätze auf drei und mehr Dimensionen erweitern, wodurch die Komplexität der zu lösenden Probleme allerdings erheblich ansteigt.

Für die Realisierung einer fünfdimensionalen Zugriffsstruktur bestehen daher zwei Alternativen: Die eine Möglichkeit besteht in der Suche nach einer Lösung der Probleme bei der Manipulation einer fünfdimensionalen Datenstruktur.

Demgegenüber besteht der in dieser Arbeit verfolgte Ansatz in der Entwicklung einer Datenstruktur, die diese Schwierigkeiten trotz eines fünfdimensionalen Datenzugriffs umgeht.

### 3.3 Konzeptuelles Modell

Im folgenden werden zunächst die räumlichen Dimensionen, die Dimension der Zeit und die Dimension der atmosphärischen Parameter, danach die benötigten syntaktischen und semantischen Metadaten betrachtet.

Zur Modellierung räumlicher Daten existiert eine Vielzahl von Strukturen. Grundsätzlich sind zwei Probleme zu lösen [FCF92]:

- Zerteilung des gesamten Raums in Teilbereiche
- Zugriff auf diejenigen Teilbereiche, die sich mit einem gegebenen Bereich des Raums schneiden, und auf die darin enthaltenen Daten.

Eine sinnvolle Zerteilung des Raums für das Datenmodell des VMS hat folgende Eigenschaften:

- Disjunktheit, was bedeutet, daß sich Gebiete nicht überlappen dürfen. Dies liegt in dem ohnehin riesigen Datenaufkommen begründet.
- Atomare (durch einen Quader beschreibbare) Gebiete. Dies vereinfacht den Zugriff auf die Daten erheblich.
- Vollständigkeit, d. h. der gesamte Raum wird von der Datenstruktur überdeckt. Diese Eigenschaft ist sinnvoll unter der Annahme, daß in allen Bereichen des betrachteten Raums auch tatsächlich Daten zu verwalten sind.

Zusätzlich besteht die Anforderung, daß die Auflösung zur Visualisierung unterschiedlicher Phänomene dynamisch eingestellt werden soll.

Die Dimension der Zeit wird modelliert durch diskrete Zeitpunkte in unterschiedlicher Granularität. Diese Zeitpunkte sind total geordnet, Zeitabstand und Zeitdauer sind durch Metriken erfaßbar [TCS<sup>+</sup>93].

Wenig Freiheitsgrade bietet die Dimension der atmosphärischen Parameter. Diese dient dazu, den atmosphärischen Parameter anzugeben, auf den sich ein Meßwert bezieht [DW89].

Die syntaktischen Meta-Daten enthalten Daten zum Informationssystem selbst. Dazu gehören strukturelle Definitionen von Daten sowie von semantischen Meta-Daten. Die syntaktischen Meta-Daten sind vom verwendeten Datenmodell abhängig.

Sie bestehen aus der Instanziierung des konzeptuellen Modells und weiteren implementierungsspezifischen Daten (Zugriffspfade, Seitenzuordnungen, ...). Für das objektorientierte Modell enthalten die syntaktischen Meta-Daten beispielsweise Klassendefinitionen.

Syntaktische Meta-Daten dienen daher sowohl dem effizienten Datenzugriff als auch der Erklärung der Daten. Diese werden beim Entwurf der Datenbank festgelegt und sollen hier nicht weiter betrachtet werden.

Semantische Meta-Daten bestehen aus Zusatzinformationen zu den gespeicherten Daten. Daher sind sie von der wissenschaftlichen Disziplin und der konkreten Anwendung abhängig [KBS93]. Die verschiedenen Arten vorkommender semantischer Meta-Daten sollen hier an einem Beispiel verdeutlicht werden. Eine Anfrage an ein Informationssystem zur Beobachtung atmosphärischer Phänomene könnte beispielsweise lauten:

*In welcher Konzentration lag Ozon über der Arktis in 30 - 40 km Höhe im Januar 1995 vor?*

Damit eine Abarbeitung solcher intuitiver, unscharfer und natürlichsprachiger Anfragen erfolgen kann, ist die Einbindung kartographischer Modelle sowie die Angabe von Namen, Einheiten und Wertebereichen notwendig.

Typ	Beispiel	Art
Katalogdaten	t+100 -> Segment A23D7655	syntaktisch
Namen	Ozon=O3, ...	semantisch
Einheiten	ppv, Grad Celsius	semantisch
Wertebereiche	1 - 2000000	semantisch
Kartographisches Modell	Arktis = (...)	semantisch
...		

Abbildung 4: Beispiele für Metadaten

Das in diesem Abschnitt entwickelte konzeptuelle Modell liefert die Grundlage für die in den beiden folgenden Kapiteln beschriebenen Studien und Experimente bezüglich verschiedener Realisierungsszenarien.

Ziel dieser Vergleichsstudie ist einerseits eine Bewertung der Möglichkeiten unterschiedlicher Systeme für die Realisierung der räumlichen Datenbank im Hinblick auf die erarbeitete Integrationsproblematik und andererseits die Entwicklung von Alternativen zur Realisierung der räumlichen Datenbank mit den bekannten fünfdimensionalen Datenstrukturen.

## 4 Realisierung durch ein Implementierungsmodell

### 4.1 Relationale DBMS

#### 4.1.1 Datenmodellierung

Im relationalen Modell werden Datenbestände durch Relationen repräsentiert. Relationen sind Mengen von gleichartig strukturierten Tupeln. Der Zugriff auf die in diesen Tabellen enthaltenen Informationen muß nach den Gesetzen der relationalen Algebra erfolgen. Diese bietet als Operationen Vereinigung, Differenz, Projektion, Selektion und Kartesisches Produkt an.

Die Datenstruktur für die SDB könnte folgendermaßen aussehen:

Relation Ausschnitt-i;					
Laenge x	Breite y	Hoehe z	Zeit t	Atm. Param. p	Messwert m
34	115	9	95031	O3	127

Abbildung 5: Beispielmodellierung für das relationale Modell

Der gesamte Datenraum würde in den Tabellen Ausschnitt-1, ... , Ausschnitt-n abgespeichert.

### 4.1.2 Zugriff auf die Daten

Ein sehr problematischer Aspekt bei der Realisierung der räumlichen Datenbank mit einem relationalen DBMS sind die Zugriffspfade. Wird der Zugriff durch mehrere eindimensionale Indizes unterstützt, so sind ständig Schnittmengenbildungen erforderlich. Dies sind allerdings sehr teure mengenalgebraische Operationen [LMK94].

Eine weitere Möglichkeit sind die kombinierten Indizes. Diese entstehen durch Anlegen mehrerer eindimensionaler Indizes auf unterschiedlichen Permutationen eines mehrdimensionalen Schlüssels. Die dadurch erreichte hohe Redundanz ist bei großen Datenmengen allerdings nicht akzeptabel.

Als Alternative kommen mehrdimensionale Hash-Strukturen (z.B. grid files) und mehrdimensionale Baum-Strukturen (z.B. octrees) in Betracht. Allerdings ist die satzorientierte Schnittstelle, die für das Anlegen von Zugriffspfaden verantwortlich ist, für die Benutzer existierender Datenbanksysteme normalerweise verborgen. Das bedeutet, daß das Anlegen mehrdimensionaler Zugriffspfade nur über ein Application Programming Interface (API) realisierbar ist [LS89].

Die Realisierung mit einem relationalen DBMS birgt noch eine Reihe weiterer Probleme, die allerdings durch spezifische Erweiterungen im Bereich der Anfragesprachen zu lösen wären:

- Die relationale Algebra bietet keine Funktionen für numerische Berechnungen an, was für die Extrapolation von Daten aber notwendig wäre.
- Die Modellierung komplexer Strukturen wird nur unzureichend unterstützt. Aggregation, Generalisierung und Rekursion werden aber zur Modellierung hierarchischer Datenstrukturen und zur Gruppierung von Objekten benötigt.
- Es gibt keine Möglichkeit für die Abarbeitung rekursiver Anfragen.

### 4.1.3 Integrationsaspekte

Wird die räumliche Datenbank (SDB) mit Hilfe eines Datenbanksystems eingerichtet, so ist ein Transformationsprozessor für die Umwandlung von Befehlen zwischen Komponentenschema und lokalem Schema notwendig.

Die Realisierung eines solchen Prozessors ist relativ aufwendig, da für eine Transformation in ein objektorientiertes Komponentenschema aus dem relationalen (lokalen) Schema Objekte so-

wie deren Objektidentität, Objektverweise (Navigation), Subtypisierung (Vererbung), Tupel-, Mengen-, Listenkonstruktion und Verhalten abgeleitet werden sollten.

Zur Transformation von Relationen zu Objekten gibt es einige komplizierte Regeln, ebenso für die Gewinnung von Objektidentitäten sowie die Tupel- und Mengen-Konstruktion.

Schwierig wird allerdings die Erzeugung von Objektverweisen und die Subtypisierung, unmöglich ist die Ableitung von Verhalten sowie die Listenkonstruktion [LL94].

Problematisch ist insbesondere die Integration zwischen räumlicher Datenbank und Visualisierungssystem:

- Eine Zerteilung des Datenraums in direkt zu visualisierende Dateien ist wohl nicht möglich, da die Dateischnittstelle von DBMS normalerweise nicht vom Benutzer manipuliert werden kann.
- Eine effiziente Abarbeitung von Benutzeranfragen, die große Datenmengen spezifizieren, ist aufgrund der beschränkten Indizierungsmöglichkeiten wohl kaum möglich.
- Welche Visualisierungssysteme unterstützt werden, hängt vom API ab.
- Hardware-Heterogenität und Verteilung dürften bei der Realisierung der SDB mit einem relationalen DBMS keine Probleme bereiten.
- Der Austausch von Daten mit anderen Wissenschaftlern ist unter Umständen schwierig, da die Beschreibungsdaten in einer Meta-Datenbasis gehalten werden und somit die Dateien nicht selbstbeschreibend sind.

## 4.2 Objektorientierte DBMS

### 4.2.1 Datenmodellierung

Das objektorientierte Modell stellt zur Datenmodellierung sehr mächtige Mechanismen zur Verfügung. Alle Daten werden in Form von Objekten gespeichert. Ein Objekt ist gekennzeichnet durch Struktur und Verhalten, welche durch die Klasse festgelegt werden.

Die Objektstruktur wird durch die Attribute des Objekts festgelegt. Die aktuelle Belegung dieser Attribute bestimmt den Objektzustand. Jedes Objekt besitzt eine eindeutige, vom Objektzustand unabhängige Identität.

Neben atomaren Typen können die Attribute eines Objekts Referenzen auf andere Objekte (Unterobjekte) enthalten, wodurch auch die Aggregation von Objekten unterstützt wird. Zur Konstruktion komplexer Typen gibt es Tupel-, Mengen- und Listenkonstruktoren.

Weitere vom objektorientierten Modell unterstützte Mechanismen sind Vererbung (Weitergabe sämtlicher struktureller und verhaltensmäßiger Eigenschaften an Untertypen) und Generalisierung (Verallgemeinerung der Eigenschaften von Untertypen). Durch sie lassen sich Klassen in einem gerichteten azyklischen Graphen anordnen.

Bei Verwendung des objektorientierten Modells wäre für die SDB folgende Modellierungsart denkbar: Die Speicherung des Datenraums erfolgt mit Hilfe einer Klasse Region.

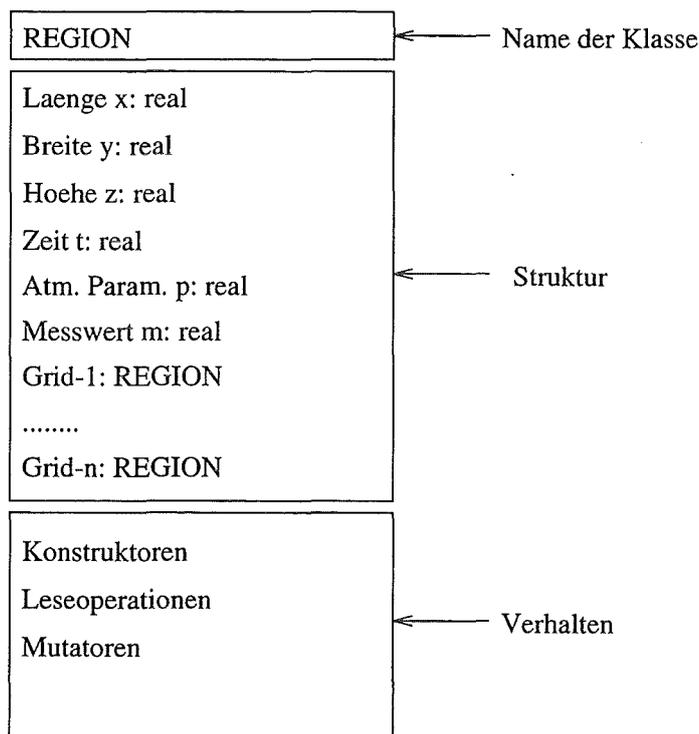


Abbildung 6: Beispielmodellierung für das objektorientierte Modell

Die Attribute Grid-1, ..., Grid-n stellen Verweise auf Unterregionen dar, die selbst wieder der Klasse Region angehören. Für  $n=8$  wäre hierbei eine Unterteilung, wie sie vom Octree her

bekannt ist, denkbar. Durch die dadurch erzeugte Verbindung der Objekte untereinander läßt sich das Problem der dynamischen Auflösung beheben.

#### 4.2.2 Zugriff auf die Daten

Beim objektorientierten Modell existiert kein vorgegebener Anfragemechanismus, d.h. das gesamte Verhalten der Objekte muß definiert werden. Im Bereich der Anfragesprachen existiert für OODBMS das Problem, daß alle auf Mengen ausgerichteten Operationen imperativ ausprogrammiert werden müssen. Alle Operatoren sind monomorph. Der Umgang mit Aggregationen erfolgt durch Navigieren, was sehr aufwendig ist.

Die grundlegende Dateiorganisationsform ist durch die Speicherstruktur der Klassen bereits festgelegt. Trotzdem ist es möglich, zusätzliche Zugriffspfade anzulegen. Dafür existieren nach [Heu92] der

- Klassenhierarchie-Index
- Equality Index (Unterstützung von Standard-Datentyp-Attributen)
- Identity Index (Unterstützung von objektwertigen Attributen)
- Pfad- oder Multi-Index (Unterstützung eines Pfades aus Klasse-Komponentenklasse- Beziehungen).

Außerdem stellen viele objektorientierte DBMS einen Versionierungsmechanismus zur Verfügung. Hiermit läßt sich die Veränderung eines Objekts im Laufe der Zeit modellieren. Der Zugriff auf die einzelnen Objekte ist durch die Struktur des entstehenden Objektgraphen festgelegt.

Jedoch bestehen bei OODBMS erhebliche Effizienzprobleme beim Lesen von Ausschnitten aus großen Punktmengen. Da sich nach [EGH<sup>+</sup>93] mit systemseitig angebotenen Indizes kaum Geschwindigkeitsgewinne feststellen lassen, ist der Einsatz spezieller räumlicher Zugriffsverfahren (Octrees,..) bei großen, mehrdimensionalen Datenmengen unabdingbar.

Das Einlesen einer großen Punktmenge in eine solche Speicherstruktur ist aber äußerst problematisch, da ab einer gewissen Größe die Baumstruktur nicht mehr in den Hauptspeicher paßt und es zu einem Seitenflattern kommt. Daher ist bei der Realisierung der SDB durch ein OODBMS ein Eingriff in die Mechanismen der physischen Blockverwaltung erforderlich.

### 4.2.3 Integrationsaspekte

Für die Integration gilt hier dasselbe wie beim relationalen Ansatz: es wird ein Transformationsprozessor für die Umwandlung von Befehlen zwischen Komponentenschema und lokalem Schema benötigt. Die Realisierung eines Transformationsprozessors zwischen zwei objektorientierten Schemata ist allerdings wesentlich einfacher.

Wie bei relationalen DBMS ist insbesondere die Integration zwischen räumlicher Datenbank und Visualisierungssystem problematisch:

- Eine Zerteilung des Datenraums in direkt zu visualisierende Dateien ist wohl nicht möglich, da die Dateischnittstelle von DBMS normalerweise nicht vom Benutzer manipuliert werden kann.
- Eine effiziente Abarbeitung von Benutzeranfragen, die große Datenmengen spezifizieren, ist aufgrund der beschränkten Indizierungsmöglichkeiten wohl kaum möglich.
- Welche Visualisierungssysteme unterstützt werden, hängt vom API ab.
- Hardware-Heterogenität und Verteilung dürften bei der Realisierung der SDB mit einem objektorientierten DBMS keine Probleme bereiten.
- Der Austausch von Daten mit anderen Wissenschaftlern ist unter Umständen schwierig, da die Beschreibungsdaten in einer Meta-Datenbasis gehalten werden und somit die Dateien nicht selbstbeschreibend sind.

## 4.3 Bewertung

Das relationale Modell hat für die spezielle Anwendung einige entscheidende Nachteile: Das Anlegen mehrdimensionaler Zugriffspfade ist nur über ein API möglich. Es gibt keine Möglichkeit für numerische Berechnungen, rekursive Anfragen und hierarchische Datenmodellierung. Dies ist aber insbesondere wegen der Forderung nach variierender Auflösung und effizientem Zugriff auf Regionen des Datenraums wichtig [B<sup>+</sup>90].

Beim objektorientierten Modell bestehen gute Möglichkeiten zur Datenmodellierung. Äußerst schwierig ist jedoch auch beim objektorientierten Modell die Gewährleistung eines effizienten Zugriffs auf Datenregionen. Problematisch ist also bei beiden Modellen, daß ein effizienter Datenzugriff bei der Abarbeitung von Benutzeranfragen wohl kaum möglich ist.

Während die Integration der Visualisierungskomponente ins Gesamtsystem über einen Transformationsprozessor realisierbar ist, kann die Integration zwischen räumlicher Datenbank und Visualisierungssystem bezüglich mehrerer Aspekte nicht befriedigend gelöst werden.

Eine Zerteilung des Datenraums in direkt zu visualisierende Dateien ist nicht möglich. Welche Visualisierungssysteme unterstützt werden, hängt vom API ab. Der Austausch von Daten ist schwierig, da diese nicht selbstbeschreibend sind.

Sowohl relationale als auch objektorientierte Datenbanksysteme haben aber den Vorteil, daß sie einen informationsorientierten Zugriff auf den Datenraum bieten.

## 5 Realisierung durch ein physikalisches Modell

### 5.1 Wissenschaftliche Datenformate

Was sind überhaupt wissenschaftliche Datenformate und inwiefern unterscheiden sie sich von Datenbanksystemen?

Wie in Abschnitt 2.4 erläutert wurde, lassen sich drei Arten von Datenmodellen unterscheiden: physikalische, Implementierungs- und konzeptuelle Modelle. Wissenschaftliche Datenformate (Scientific Data Formats - SDF) bieten im Gegensatz zu Datenbanksystemen lediglich ein physikalisches Datenmodell an.

Da die von Implementierungsmodellen angebotenen Modellierungsmöglichkeiten nur sehr spärlich genutzt werden können, ist zu überlegen, ob diese Art des Low-Level-Modellings für das Datenmodell der SDB angemessener ist als die Darstellung durch ein Implementierungsmodell.

Wissenschaftliche Datenformate dienen nicht nur zur Haltung, sondern vor allem auch zum Austausch wissenschaftlicher Daten, d.h. es werden Hardwareunabhängigkeit und Import/Export-Möglichkeiten unterstützt.

Die bearbeiteten wissenschaftlichen Datenformate sind CDF, netCDF, HDF, FITS, GRIB, VICAR, PDS, SAIF, SDTS und HDS. Weiter betrachtet werden nur HDF und netCDF, da bei CDF keine Datenaggregationen adressierbar sind und HDS eine zu statische Speicherstruktur besitzt. Die anderen Formate sind ohnehin ungeeignet, da sie auf die Modellierung komplexer Objekte oder auf Bildverarbeitung ausgerichtet sind, nicht auf effizienten Zugriff auf mehrdimensionale Strukturen.

Beim Versuch, wissenschaftliche Datenformate in die Fünf-Schichten-Architektur für Datenbanksysteme [LMK94] einzuordnen, kann festgestellt werden, daß weder Mengen- noch Satz-Manager existieren. Dagegen können bei wissenschaftlichen Datenformaten der Satzspeicher-Manager und der Datei-Manager vom Benutzer direkt manipuliert werden.

Im Gegensatz zu Datenbanksystemen gibt es bei wissenschaftlichen Datenformaten kein Transaktionskonzept. Bei DBMS sind die sogenannten ACID-Transaktionsprinzipien üblich, die auf Dienstleistungen kurzer Dauer und geringen Datenvolumens zugeschnitten sind.

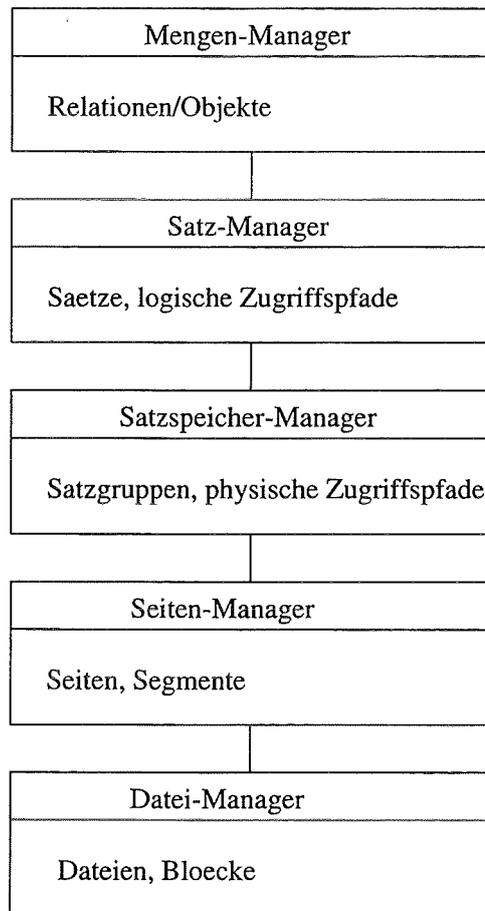


Abbildung 7: Fünf-Schichten-Architektur für Datenbanksysteme

- Atomizität: Eine Transaktion hinterläßt erst bei erfolgreichem Abschluß eine Wirkung, die dann allgemein sichtbar ist.
- Konsistenz: Der Datenbasiszustand bleibt nach Durchführung einer Transaktion konsistent.
- Isolation/Koordination: Nebenläufige Transaktionen beeinflussen sich nicht.
- Dauerhaftigkeit: Die Wirkung einer erfolgreichen Transaktion geht nicht verloren.

Wegen des Fehlens von Transaktionsprinzipien müßten deshalb bei einer Realisierung der räumlichen Datenbank mit einem wissenschaftlichen Datenformat neben Konkurrenzverhalten und Synchronisationsprotokollen auch die Verdrängungs-, Einbringungs- und Auslagerungsstrategien geklärt werden.

Von besonderer Bedeutung für das VMS ist vor allem die Frage nach der Sperrgranularität: Bei kleinen Sperreinheiten müssen enorm viele Sperren verwaltet werden. Dagegen besteht bei großen Sperreinheiten die Gefahr, daß die Parallelität von Transaktionen stark reduziert wird [Moe94].

## 5.2 HDF

### 5.2.1 Einführung in HDF

HDF (Hierarchical Data Format) ist ein wissenschaftliches Datenformat, das für die Haltung und den Austausch graphischer und numerischer Daten entwickelt wurde.

HDF unterstützt neben Rasterbildern, Farbpaletten, Text und Binärtabellen mehrdimensionale Array-Strukturen (Scientific Data Sets - SDS). Jede Datei enthält ihre eigenen Meta-Daten. Diese einzelnen Objekte können hierarchisch gruppiert werden. HDF ist auf vielen Plattformen einsetzbar [Nat93].

Die HDF-Bibliothek besteht aus Fortran- und C-Schnittstellen auf drei verschiedenen Ebenen: Low-Level-Interfaces, Single-and-Multi-File-Interfaces sowie Utilities.

Die Low-Level-Interfaces dienen zur Datei- und Speicherverwaltung sowie zur Fehlerbehandlung. Die Single-and-Multi-File-Interfaces beziehen sich auf die einzelnen Datentypen. Das DFSD/SD-

Interface unterstützt Scientific Data Sets. Die Utilities arbeiten auf bestehenden HDF-Dateien und dienen beispielsweise zur Gruppierung von Objekten.

### 5.2.2 Datenmodellierung

Ein Scientific Data Set enthält

- die Dimensionen und die Anzahl der Dimensionen
- den Datentyp
- die aktuellen Datenwerte

sowie optional Interpretationsdaten

- Skalen für die verschiedenen Dimensionen
- Maximum, Minimum
- Labels für Dimensionen und Daten
- Einheiten für Dimensionen und Daten
- Koordinatensysteme
- Kalibrierungsinformationen
- Füllwerte für fehlende Daten
- Formatspezifikationen zum Verbreiten von Daten oder Dimensionen.

Von der High-Level-Library NCSA HDF UCD wird zusätzlich das Vset-Konzept als Strukturierungsmöglichkeit angeboten. Vsets bestehen aus Vgroups und Vdata. Vdata enthalten Daten, Vgroup Verweise auf andere Elemente (Vdatas, Vgroups, SDS).

Eine Realisierung der räumlichen Datenbank könnte mit Scientific Data Sets erfolgen. In fünfdimensionalen Arrays wird den Dimensionen Länge x, Breite y, Höhe z, Zeit t, atmosphärischer Parameter p ein Meßwert m zugeordnet.

Durch das Vset-Konzept läßt sich eine hierarchische Speicherstruktur verwirklichen. Diese kann ähnlich zu der beim objektorientierten Modell beschriebenen Struktur gestaltet werden.

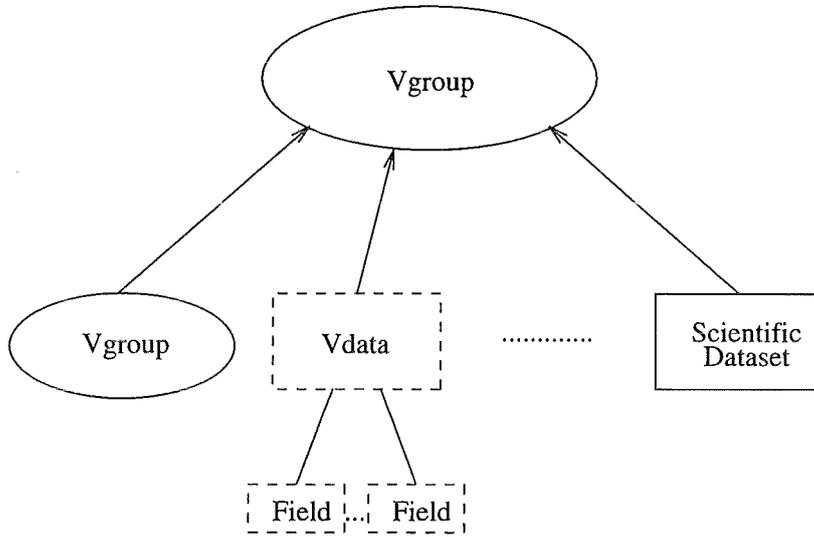


Abbildung 8: Struktur einer Vgroup

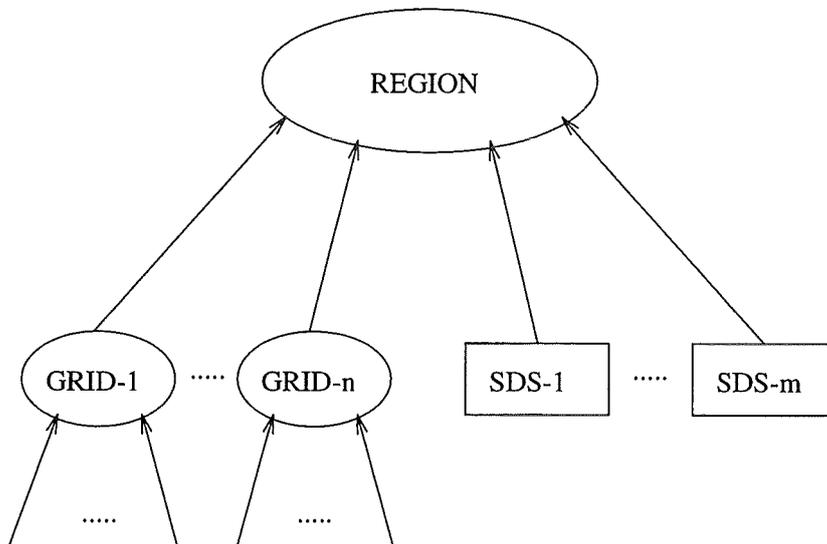


Abbildung 9: Beispielmodellierung mit HDF

### 5.2.3 Zugriff auf die Daten

Das DFSD/SD-Interface enthält Funktionen zum Lesen und Schreiben von Scientific Data Sets sowie zum Arbeiten mit Dimensionen und Attributen.

Die High-Level-Library NCSA HDF UCD dient zur Speicherung und Verarbeitung dreidimensionaler Daten, Erzeugung polygonaler Konnektivitätslisten, zur Konsistenzerzeugung zwischen benutzererzeugten Konnektivitätssequenzen und Defaultsequenzen sowie zur Speicherung von verwandten Arraydaten in Vsets für die hierarchische Gruppierung von Daten.

Beim Experimentieren mit HDF hat sich allerdings herausgestellt, daß sich das Vset-Konzept nur innerhalb von Dateien anwenden läßt. Während der Zugriff auf Daten innerhalb spezifizierter Dateien daher sehr gut möglich ist, existieren keinerlei HDF-Funktionen zur Auswahl von Dateien.

HDF ermöglicht es also, den Datenraum in direkt zu visualisierende Dateien zu zerlegen, die Übergabe dieser Dateien an ein Visualisierungssystem müßte jedoch von einem Anwendungsprogramm übernommen werden.

### 5.2.4 Integrationsaspekte

Mit Hilfe der HDF-Bibliothek kann ein Zugriffsprozessor erstellt werden, der die Integration zwischen KBMS und VMS realisiert.

Für die Integration von räumlicher Datenbank und Visualisierungssystem gilt:

- Der Datenraum kann in einzelne, direkt zu visualisierende Dateien zerteilt werden. Eine effiziente Abarbeitung von Benutzeranfragen, die große Datenmengen spezifizieren, ist allerdings nicht möglich, da HDF-Funktionen keinen informationsorientierten Zugriff auf die Dateien bieten können.
- Viele Visualisierungssysteme können HDF-Dateien direkt oder über Konvertierungsprogramme (DDI,...) einlesen. Diese unterstützen jedoch üblicherweise nur drei Dimensionen.
- Hardware-Heterogenität und Verteilung werden hervorragend unterstützt. Der Austausch von Daten mit anderen Wissenschaftlern ist aufgrund selbstbeschreibender Dateien sehr gut möglich.

## 5.3 NetCDF

### 5.3.1 Einführung in netCDF

NetCDF (NETwork Common Data Format) ist ein wissenschaftliches Datenformat für die Haltung und den Austausch numerischer Daten, welches insbesondere für die Atmosphärenforschung sehr verbreitet ist.

NetCDF-Dateien sind selbstbeschreibend, d.h. sie enthalten ihre eigenen Meta-Daten. NetCDF ist auf vielen Plattformen einsetzbar. Mit Hilfe eines hyperslab-Konzepts können kleine Subsets eines großen Datensets effizient gespeichert und wiedergefunden werden. Zugriff auf eine netCDF-Datei können ein Schreiber und mehrere Leser gleichzeitig haben.

Die netCDF-Bibliothek besteht aus Fortran- und C-Interfaces. Außerdem existiert eine Prototyp-Version einer C++-Schnittstelle [RDE93].

### 5.3.2 Datenmodellierung

Eine netCDF-Datei besteht aus Dimensionen, Variablen und Attributen. Dimensionen haben einen Namen und eine Größe. Sie dienen dazu,

- Arten und Größen von Variablen zu spezifizieren
- Variablen, die auf einem gewöhnlichen grid definiert sind, zu identifizieren und in Beziehung zu setzen
- Koordinatensysteme zu erstellen.

Variablen sind mehrdimensionale Arrays desselben Typs. Eine Variable hat einen Namen, einen Datentyp und eine Form, die durch eine Liste von Dimensionen spezifiziert wird.

Attribute enthalten Meta-Informationen.

Die einzige Datenstruktur, die von netCDF direkt unterstützt wird, ist eine Sammlung von skalaren oder mehrdimensionalen Variablen mit zugehörigen Vektorattributen.

Da Datenstrukturen, die Zeiger enthalten, nicht direkt unterstützt werden, ist eine hierarchische Modellierung des Datenraums nur schlecht möglich. Die Modellierung erfolgt nach dem Schema in Abbildung 10.

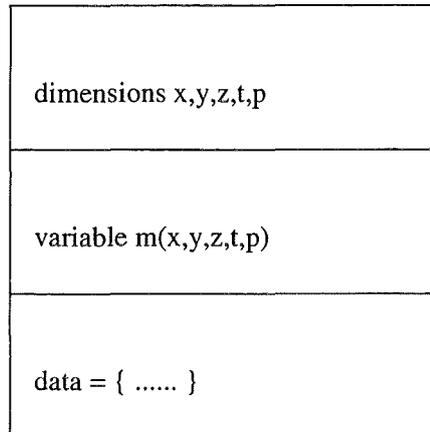


Abbildung 10: Beispielmmodellierung mit netCDF

### 5.3.3 Zugriff auf die Daten

Die netCDF-Bibliothek bietet Funktionen zum Erzeugen, Lesen und Schreiben von Dateien, zum Arbeiten mit Dimensionen, Variablen, Attributen und Daten sowie zur Fehlerbehandlung.

Der Zugriff auf die Daten kann über hyperslabs erfolgen. Ein hyperslab ist ein Teil einer mehrdimensionalen Variable, der durch Angabe eines Ankerpunkts und einer Liste von Kantenlängen bezüglich der Dimensionen der Variable spezifiziert wird. Dadurch lassen sich Teilbereiche des Datenraums effizient adressieren.

NetCDF ermöglicht es genauso wie HDF, den Datenraum in direkt zu visualisierende Dateien zu zerlegen. Es existieren jedoch ebenfalls keinerlei netCDF-Funktionen, mit denen die Auswahl von Dateien zur Übergabe an ein Visualisierungssystem vorgenommen werden könnte.

### 5.3.4 Integrationsaspekte

Mit Hilfe der netCDF-Bibliothek kann ein Zugriffsprozessor erstellt werden, der die Integration zwischen KBMS und VMS realisiert. Die Realisierung dieses Prozessors dürfte bei Verwendung des C++-Interfaces besonders leichtfallen, da das KBMS mit einem OODBMS realisiert werden soll.

Für die Integration von räumlicher Datenbank und Visualisierungssystem gilt ähnliches wie bei HDF:

- Der Datenraum kann in einzelne, direkt zu visualisierende Dateien zerteilt werden. Eine effiziente Abarbeitung von Benutzeranfragen, die große Datenmengen spezifizieren, ist allerdings mit netCDF-Funktionen nicht möglich, da keine informationsorientierte Adressierung von Dateien geboten werden kann.
- Viele Visualisierungssysteme können netCDF-Dateien direkt oder über Konvertierungsprogramme (DDI,...) einlesen. Diese unterstützen jedoch üblicherweise nur drei Dimensionen.
- Für die Ausgabe von Daten an ein Visualisierungssystem sollen demnächst Operatoren zur Vereinigung von netCDF-Dateien sowie zur Erzeugung einer anderen Ausgabe (z.B. Text oder Graphik) aus netCDF-Dateien und umgekehrt zur Erzeugung von netCDF-Dateien aus Text-Dateien existieren.
- Hardware-Heterogenität und Verteilung werden hervorragend unterstützt. Der Austausch von Daten mit anderen Wissenschaftlern ist aufgrund selbstbeschreibender Dateien sehr gut möglich.

## 5.4 Bewertung

Der entscheidende Nachteil wissenschaftlicher Datenformate für die Realisierung der räumlichen Datenbank ist die Tatsache, daß sie keinen informationsorientierten Zugriff auf den Datenraum bieten, da keinerlei Funktionalität zur Verwaltung und Auswahl von Dateien zur Übergabe an ein Visualisierungssystem vorhanden ist.

Sie ermöglichen aber die Realisierung einer Datenstruktur, die innerhalb spezifizierter Dateien einen sehr effizienten, informationsorientierten Datenzugriff bietet. Daneben werden SDF-Dateien von vielen Visualisierungssystemen als Eingabedaten akzeptiert.

Wissenschaftliche Datenformate unterstützen die enge Kopplung mit einem Visualisierungssystem durch die Möglichkeit zur expliziten Zerteilung des Datenraums in direkt ans Visualisierungssystem zu übergebende Dateien. Allerdings dürfen die Eingabedaten an Visualisierungssysteme nur von drei Dimensionen abhängen.

Über die vorhandenen Bibliotheken kann ein Zugriffsprozessor zur Integration der Visualisierungskomponente ins Gesamtsystem erfolgen. Die Kopplung an ein Visualisierungssystem kann ebenfalls mit Hilfe dieser Bibliotheken und mit Konvertierungsroutinen erfolgen.

Außerdem werden Hardwareunabhängigkeit und Import-/Exportmöglichkeiten hervorragend unterstützt. Wissenschaftliche Datenformate besitzen keine Transaktionsmechanismen und keine Satz- und Mengenmanager wie Datenbanksysteme.

Fraglich ist auch, ob die anfallenden Meta-Daten angemessen verwaltet werden können. Die Angabe von Skalen, Einheiten und Wertebereichen ist möglich, eine Einbindung kartographischer Modelle etc. dürfte aber Schwierigkeiten bereiten.

## 6 Alternative: Hybride Lösung

### 6.1 Konzept einer hybriden Lösung

#### 6.1.1 Architektur

Eine hybride Lösung des Entwurfsproblems könnte in der Realisierung der räumlichen Datenbank sowohl mit einem wissenschaftlichen Datenformat als auch mit einem objektorientierten DBS erreicht werden.

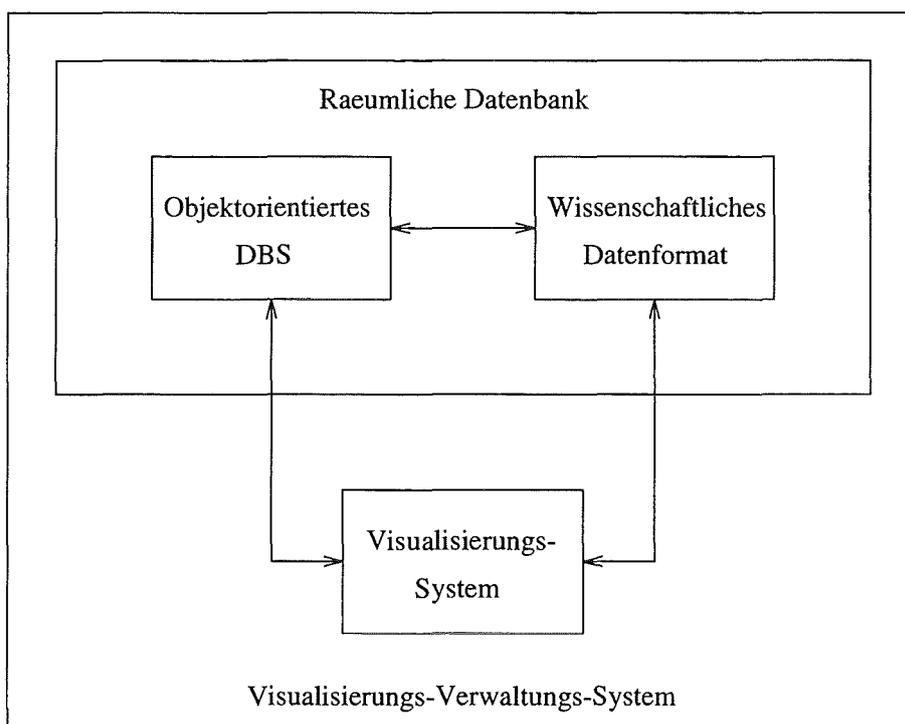


Abbildung 11: Hybride Modellierung des VMS

Das Ziel dieser Lösung ist, für die konkrete Problemstellung die Vorteile von Datenbanksystemen mit den Vorteilen von wissenschaftlichen Datenformaten zu vereinen und die Beschränkungen

von Visualisierungssystemen bezüglich der Eingabedaten zu umgehen.

Wissenschaftliche Datenformate erlauben eine explizite Zerteilung des Datenraums in direkt ans Visualisierungssystem zu übergebende Dateien, eine einfache Integration der räumlichen Datenbank mit einem Visualisierungssystem, die Verwendung verschiedener Hardwareumgebungen und Import/Export von Daten.

Ein objektorientiertes DBMS dagegen bietet die Möglichkeit, die Meta-Daten der durch das wissenschaftliche Datenformat erstellten Dateien zu verwalten, und stellt Transaktionsmechanismen zur Verfügung.

Die innovative Idee bei diesem Ansatz ist also die Verwendung eines Datenbankssystems zum informationsorientierten Zugriff auf Daten, die von einem wissenschaftlichen Datenformat verwaltet werden.

Der hybride Ansatz eröffnet die Perspektive, die Komplexität der Probleme sowohl bei der Manipulation der fünfdimensionalen Datenstruktur als auch bei der Integration von räumlicher Datenbank und Visualisierungssystem durch die Aufteilung der Dimensionen auf Datenbanksystem und wissenschaftliches Datenformat erheblich zu reduzieren, obwohl ein fünfdimensionaler Datenzugriff gewährleistet wird.

Zu entwickeln ist einerseits die Datenhaltungsstrategie, also die Aufteilung der Zuständigkeiten auf objektorientiertes DBMS und wissenschaftliches Datenformat bezüglich der Verwaltung der fünf Dimensionen atmosphärische Parameter ( $p$ ), Geokoordinaten ( $x,y$ ), Höhe ( $z$ ) und Zeit ( $t$ ) sowie der Metadaten.

Andererseits ist eine Anfragebearbeitungsstrategie im Hinblick auf einen fünfdimensionalen Datenzugriff und auf die zu lösenden Schnittstellenprobleme zu gestalten, wobei eine wichtige Beschränkung darin besteht, daß die meisten Visualisierungssysteme nur maximal dreidimensionale Eingabedaten akzeptieren.

### 6.1.2 Datenhaltung

Da das OODBMS für die Abarbeitung deklarativer Anfragen zuständig ist, ist es auf jeden Fall sinnvoll, mit dem OODBMS die Metadaten, bestehend aus Katalogdaten, kartographischen Modellen usw., zu verwalten. Darüberhinaus sollte ein Teil der Dimensionen vom OODBMS verwaltet werden, weil Eingaben ans Visualisierungssystem nur von drei Dimensionen abhängig sein dürfen.

Wegen der erwähnten Vorteile bietet sich ein SDF für die Verwaltung der Meßdaten in Abhängigkeit von den drei räumlichen Dimensionen an. So können mit ausreichender Performanz

beliebige Ausschnitte aus dem dreidimensionalen Raum extrahiert werden.

Demgegenüber erscheint eine Realisierung der Dimension Zeit mit Hilfe von Versionierungsmechanismen oder Mengenkonstruktoren interessant, wie sie von OODBMS angeboten werden, da sich Messungen im Normalfall auf Folgen von Zeitpunkten oder Zeitintervallen unterschiedlicher Granularität beziehen.

Anfragen beziehen sich immer nur auf einen oder wenige atmosphärische Parameter, daher bietet es sich an, den atmosphärischen Parameter als Katalogdatum zu SDF-Dateien mit räumlichen Dimensionen und Meßwerten zu führen.

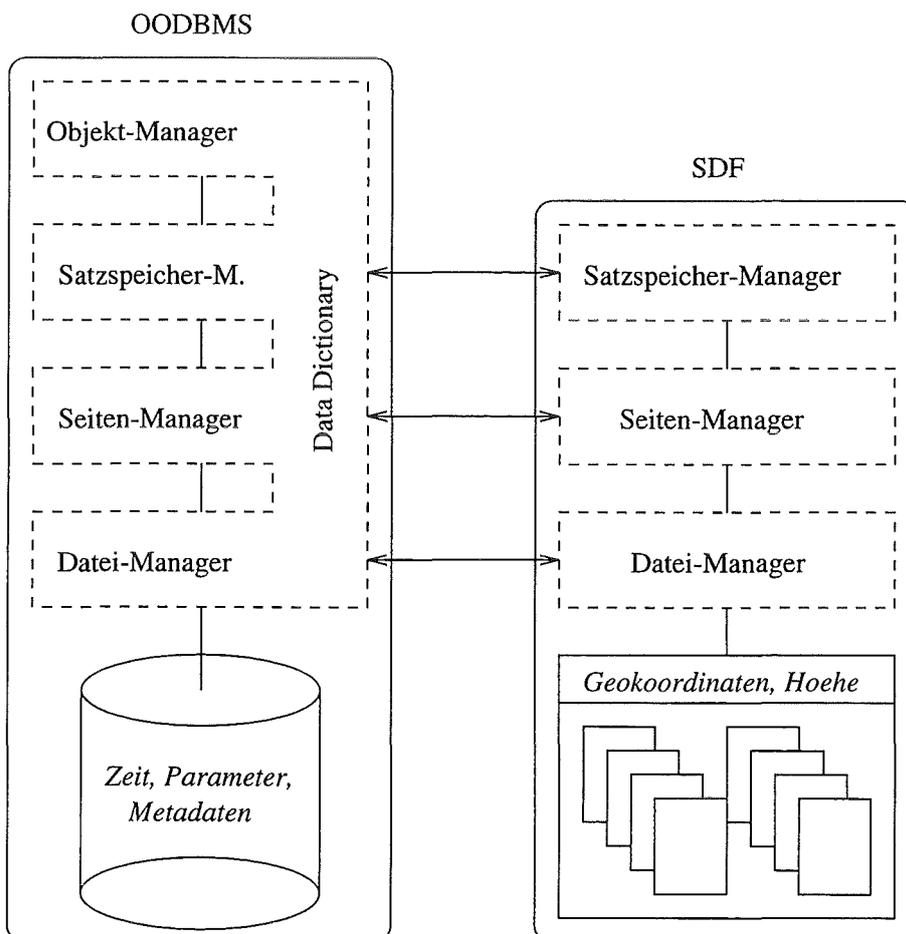


Abbildung 12: Architektur der räumlichen Datenbank

Diese Überlegungen führen zur Entwicklung eines Datenhaltungsmodells mit folgender Struktur: Mit SDF-Dateien werden Meßwerte in Abhängigkeit von den räumlichen Dimensionen verwaltet, die sich auf genau einen Zeitpunkt und genau einen Parameter beziehen. Diese Dateien werden im folgenden auch als Raumobjekte bezeichnet. Raumobjekte enthalten also zu einem Zeitstempel und einem Parameter Meßdaten für alle Geokoordinaten und Höhen.

Das OODBMS verwaltet die Metadaten dieser Raumobjekte. Dazu gehören einerseits Katalogdaten bezüglich der Dimensionen Zeit und atmosphärische Parameter sowie der Auflösung, andererseits semantische Metadaten wie Namen, Wertebereiche und kartographische Modelle.

Semantische Metadaten dienen zur Speicherung von Informationen, die die Abarbeitung unscharfer Anfragen ermöglichen. Es werden sowohl für die fünf Dimensionen als auch für Auflösung und Visualisierungsmodi Metadaten benötigt.

Für die Verwaltung der Raumobjekte durch das OODBMS müssen Mechanismen zur Verfügung gestellt werden, die einen effizienten Zugriff auf die Raumobjekte entlang der Dimensionen Zeit und atmosphärische Parameter gewährleisten.

Vom OODBMS müssen außerdem Transaktionsmechanismen zur Verfügung gestellt werden. Es bietet sich an, die Raumobjekte als Basiseinheit für Sperren zu verwenden.

Das Problem der Auflösung kann entweder durch Selektion bzw. Interpolation zur Laufzeit oder durch redundante Haltung von Daten geringerer Auflösung realisiert werden.

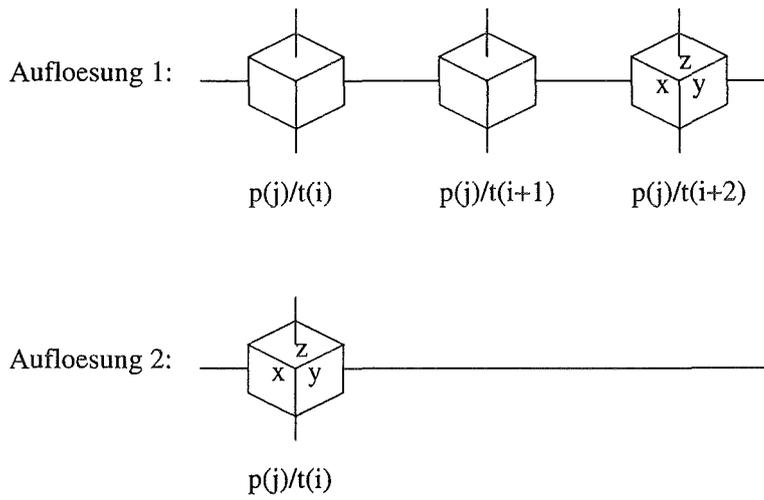
Bei redundanter Datenhaltung werden Dateien mit interpolierten Daten angelegt. Diese Dateien mit Daten geringerer räumlicher und zeitlicher Granularität werden getrennt von den übrigen Dateien verwaltet. Dadurch ist ein besonders effizienter Zugriff auf die weniger genauen Daten gewährleistet.

Für die Realisierung der geringeren Auflösung zur Laufzeit bestehen zwei Möglichkeiten. Die eine Möglichkeit ist, aus einem Teil der Dateien (zeitliche Auflösung) Stichproben aus den Daten (räumliche Auflösung) zu extrahieren. Die andere Möglichkeit besteht in der Durchführung einer Interpolation über den entsprechenden Daten zur Laufzeit, was allerdings recht zeitaufwendig und somit nicht sinnvoll ist.

### 6.1.3 Anfragebearbeitung

Deklarative Anfragen bezüglich der fünf Dimensionen werden vom Benutzer an das OODBMS gestellt. Diese Anfragen werden geparkt und durch Katalogdaten und andere Metadaten optimiert.

Realisierung mit Collections:



Realisierung mit Versionen:

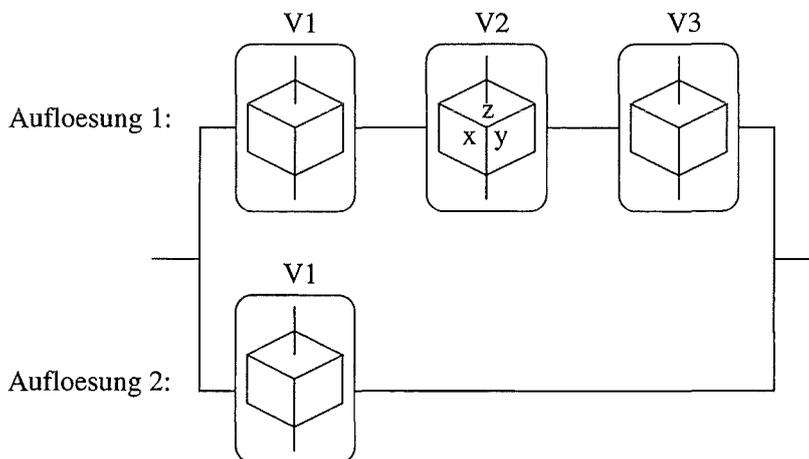


Abbildung 13: Beispielmodellierung bei hybrider Lösung

Vom OODBMS werden die jeweiligen atmosphärischen Parameter, Zeitstempel und Beschreibungsdaten geliefert. Die Übertragung der räumlichen Dimensionen und der Meßwerte erfolgt aus SDF-Dateien.

Wie funktioniert bei dieser Aufteilung der Verwaltung der Dimensionen (x,y,z,t,p) und der Metadaten auf OODBMS und SDF ein fünfdimensionaler Datenzugriff und die Abarbeitung deklarativer Anfragen?

Eine fünfdimensionale Zugriffsstruktur läßt sich durch eine Kombination aus Time Sequence Collections oder Temporal Version Graphs (Zeit und Parameter) mit den von wissenschaftlichen Datenformaten angebotenen Zugriffsstrukturen (räumliche Dimensionen) verwirklichen.

Ein Temporal Version Graph besteht aus hierarchisch geordneten temporalen Objekten oder Attributen, die als Funktionen der Zeit modelliert wurden [TCS<sup>+</sup>93].

Die Grundlage von Time Sequence Collections sind Tripel  $\langle s,t,a \rangle$  mit s als Surrogat des Objekts (z.B. Raumobjekt), t für die Zeit und a für das Attribut (z.B. atmosphärische Parameter), sogenannte Temporal Data Values, die zu total geordneten Zeitsequenzen zusammengefaßt werden.

Eine Time Sequence Collection besteht aus einem Tripel (S,T,A) mit S, T und A als Domänen von Surrogat, Zeit und Attribut [TCS<sup>+</sup>93]. Daher ist ein Zugriff auf die Objekte sowohl über die Zeit als auch das Attribut, also über Zeit und atmosphärische Parameter, möglich.

Diese fünfdimensionale Zugriffsstruktur ist die Grundlage für eine Abarbeitungsstrategie, die eine informationsorientierte Aktivierung des Visualisierungssystems erlaubt:

Der Zugriff auf die Raumobjekte erfolgt entlang der Dimensionen t und p mit Hilfe der Zugriffsmechanismen, die von Time Sequence Collections oder Temporal Version Graphs zur Verfügung gestellt werden.

Unter den Objekten der gewünschten Auflösung werden diejenigen identifiziert, die mit den durch die Benutzeranfrage spezifizierten Zeitpunkten und atmosphärischen Parametern korrespondieren. Dies bedeutet, daß zu einer gegebenen Anfrage zunächst die entsprechenden Zeitstempel und Parameter bestimmt werden. Diese bestimmen die zur Extraktion von Meßdaten in Frage kommenden Dateien.

Außerdem setzt das OODBMS mit Hilfe von Metadaten räumliche Spezifikationen wie Arktis,... in Grid-Bereiche um, wodurch die aus SDF-Dateien auszulesenden Datenbereiche bestimmt werden.

Gleichzeitig werden vom Benutzer Visualisierungsmodi und Auflösung festgelegt. Somit sind alle Informationen vorhanden, die zur Initialisierung der Konvertierungsroutine und zur Spezifikation

derjenigen SDF-Dateien, in denen die zu visualisierenden Meßdaten enthalten sind, benötigt werden.

Nach erfolgter Initialisierung mit Metadaten und Visualisierungsmodi erzeugt die Konvertierungsroutine durch Auslesen derjenigen Meßdaten, die sich auf den spezifizierten Raumausschnitt beziehen, aus SDF-Dateien eine Datei, die die zu visualisierenden Daten und Metadaten in einem vom Visualisierungssystem lesbaren Format enthält.

Hierzu wird auf den spezifizierten Raumobjekten eine Anfrage gestartet, die den Teil der Meßdaten extrahiert, der sich auf den gewünschten Ausschnitt aus dem Raum bezieht.

Diese Meßdaten werden zusammen mit Werten für die fünf Dimensionen und Angaben zur Darstellung der Daten in ein vom Visualisierungssystem lesbares Format überführt. Die somit erzeugte Datei bildet die Eingabe an das Visualisierungssystem.

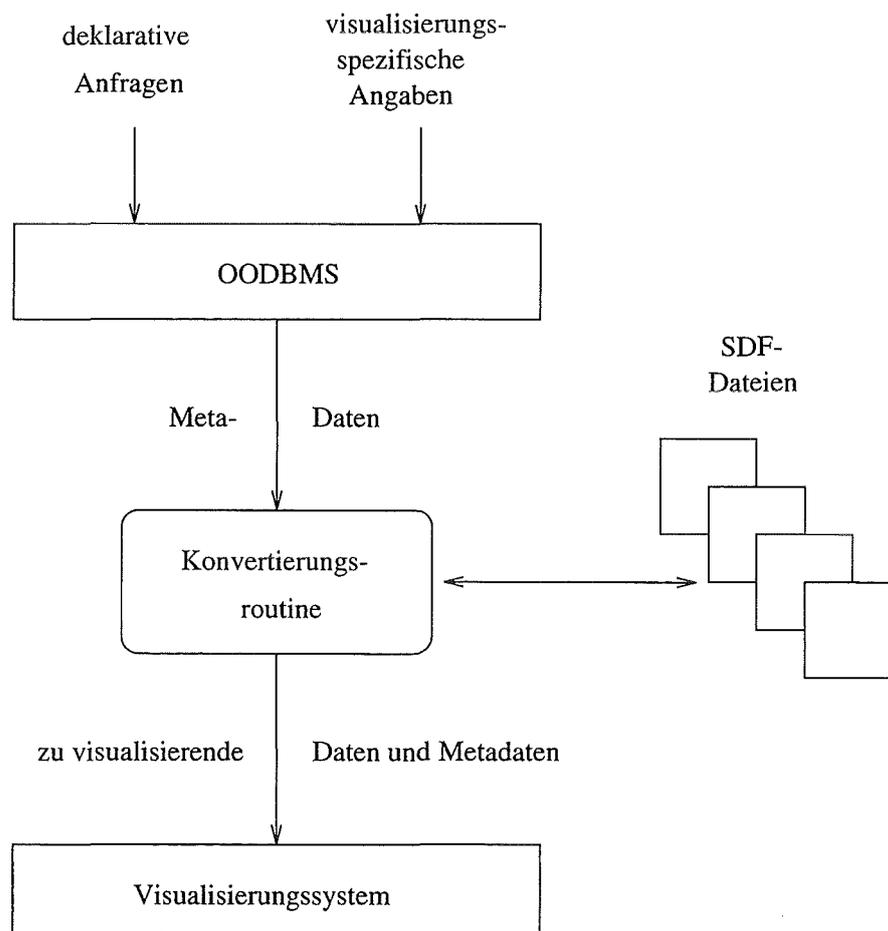


Abbildung 14: Anfragebearbeitung

## 6.2 Realisierung eines Prototyps

### 6.2.1 Aufbau der räumlichen Datenbank

Zur Realisierung der räumlichen Datenbank wurden das objektorientierte Datenbanksystem ObjectStore und das wissenschaftliche Datenformat netCDF verwendet.

Der Aufbau der räumlichen Datenbank erfolgt durch das Anlegen von Objektklassen bzw. Dateien und der Instanziierung dieser Strukturen durch Anwendungsprogramme, die auf die Methoden der Objekte bzw. auf netCDF-Funktionen zurückgreifen.

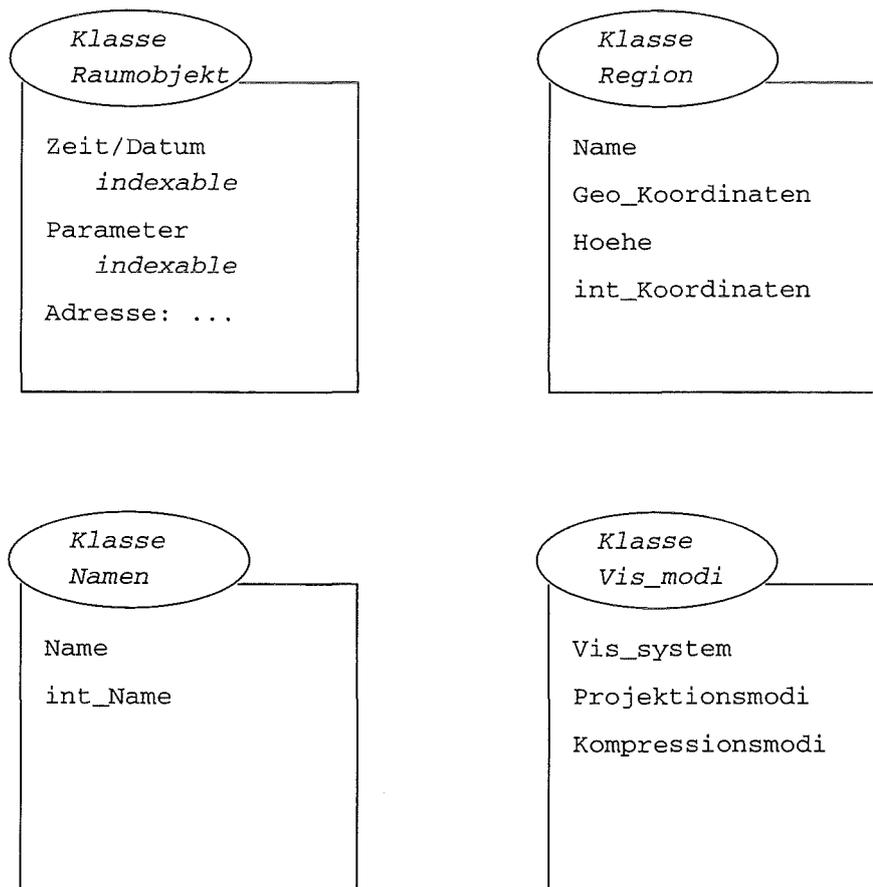


Abbildung 15: Objektklassen der räumlichen Datenbank

Die Klassen *Raumobjekt*, *Region*, *Namen* und *Vis\_modi* geben die Struktur für die mit ObjectStore anzulegenden Objekte vor.

Durch vielfache Instanziierung der Klasse *Raumobjekt* und Verwendung von Collections- oder Versionskonzepten wird eine Objektstruktur angelegt, die einen effizienten Zugriff auf die Adressen der netCDF-Raumobjekte entlang der Dimensionen Zeit und Parameter bietet.

Das Datenbanksystem ObjectStore stellt Collections (Sets, Bags, Lists) zur Verfügung, auf denen der Zugriff durch Indizes und Hashfunktionen optimiert werden kann. Die Modellierung der Dimension Zeit kann auch durch den von ObjectStore zur Verfügung gestellten Versionierungsmechanismus vorgenommen werden [Obj94].

Zur Verwaltung der netCDF-Raumobjekte wurden daher zwei Lösungsansätze erprobt:

Im einen Fall sind die einzelnen Objekte, bestehend aus Zeit/Datum-Stempel, atmosphärischem Parameter und Adresse des zugehörigen netCDF-Raumobjekts, zu einem Set zusammengefaßt. In dieser Objektmenge wird mit Hilfe von Indizes auf Zeit und Parameter gesucht.

Im anderen Fall wird die Modellierung der Zeit durch einen Versionierungsmechanismus übernommen. Für jeden Zeitpunkt wird eine Version eines Objektes für die Verwaltung der Adressen der netCDF-Raumobjekte zu diesem Zeitstempel und zu allen atmosphärischen Parametern angelegt. (Siehe auch Abbildung 16).

Das Einfügen eines neuen Raumobjektes geschieht in zwei Schritten: Zunächst wird ein Objekt der Klasse *Raumobjekt* angelegt, welches Zeitstempel und atmosphärischen Parameter sowie die Adresse der zu verwaltenden netCDF-Datei enthält.

Mit derselben ObjectStore-Anwendung wird daraufhin die korrespondierende netCDF-Datei mit den Dimensionen  $x, y, z$  und der Variablen  $m(x, y, z)$  für die Meßwerte angelegt. Sowohl Variablen als auch Dimensionen haben Attribute, darunter Zeitstempel, Datumstempel, Titel, Einheiten, Wertebereiche. Die eigentliche Beschreibung der netCDF-Datei mit Meßdaten geschieht sehr effizient durch hyperslabs, das heißt es werden Daten in einen Bereich geschrieben, der durch einen Startpunkt und durch Kantenlängen entlang der drei Dimensionen spezifiziert wird.

Die Klassen *Region*, *Namen* und *Vis\_modi* dienen zur Modellierung der semantischen Metadaten: Die Klasse *Region* gewährleistet die Umsetzung deklarativer Benutzereingaben bezüglich der räumlichen Dimensionen in eine interne Repräsentation.

Unschärfe Angaben bezüglich Zeit und atmosphärischem Parameter werden mit Hilfe der Klasse *Namen* in eine interne Repräsentation überführt. Außerdem können mit der Klasse *Vis\_modi* die Visualisierungsmodi verschiedener Visualisierungssysteme beschrieben werden.

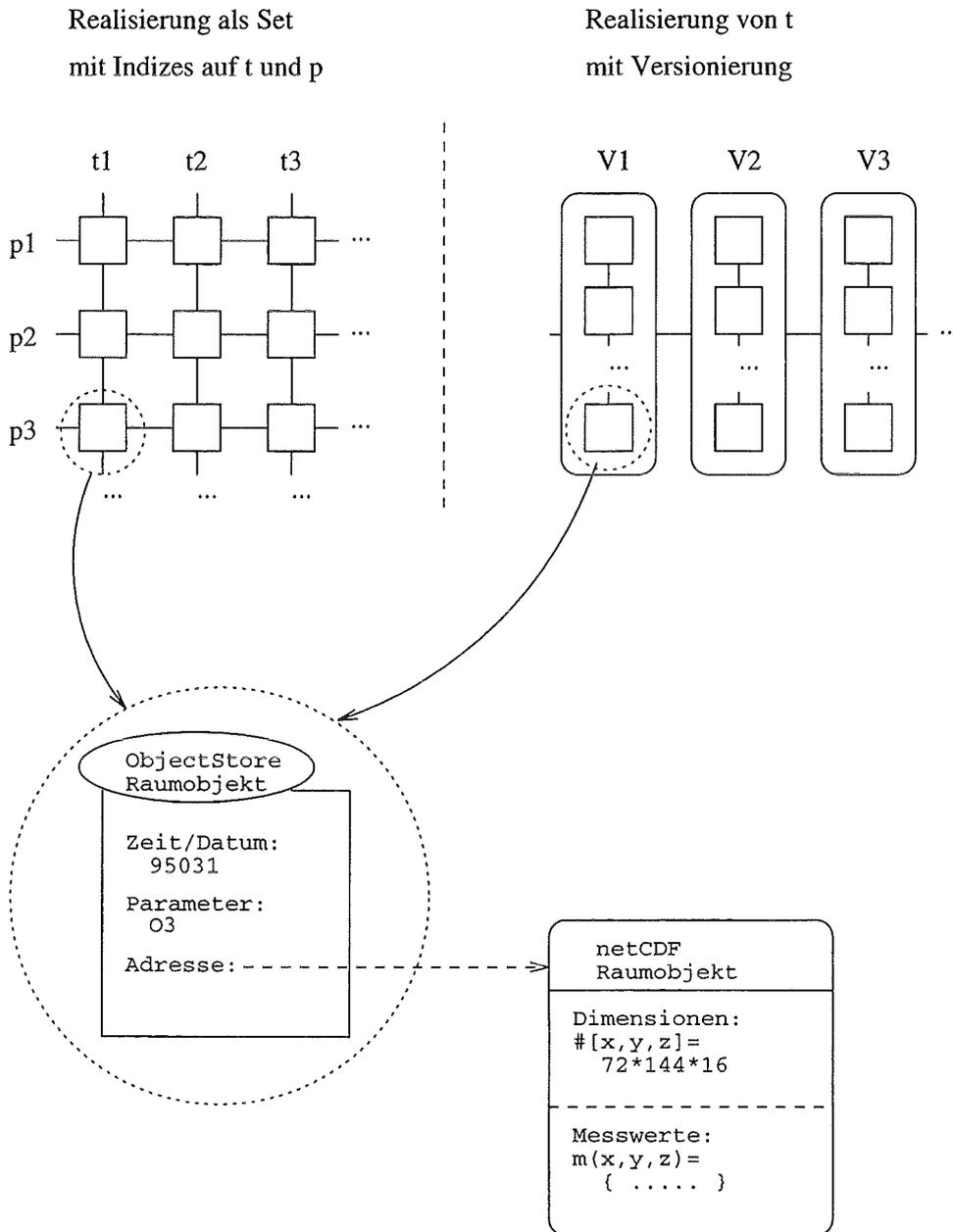


Abbildung 16: Raumobjektstruktur

Die Umsetzung der Anfrage

*In welcher Konzentration lag Ozon über der Arktis in 30 - 40 km Höhe im Januar 1995 vor?*

geschieht demnach durch die Identifikation interner Repräsentationen bezüglich der Angaben zu den fünf Dimensionen:

- Ozon  $\rightarrow$  O3
- Arktis  $\rightarrow$  [180 Grad West - 180 Grad Ost, 90 Grad Nord - 70 Grad Nord]
- Januar 1995  $\rightarrow$  [95001, ... , 95031]

### 6.2.2 Schnittstelle zum Visualisierungssystem

Als Visualisierungssystem wurde Vis5D ausgewählt, es kommen jedoch auch andere Systeme (AVS, IRIS Explorer, ...) in Betracht, da die räumliche Datenbank unabhängig vom verwendeten Visualisierungssystem gestaltet wird.

Ursprünglich wurde Vis5D zur Visualisierung numerischer Wettermodelle entwickelt. Es erfordert Eingabedaten, die auf der Basis eines Gittermodells mit drei räumlichen Dimensionen, versehen mit einem Zeit/Datum-Stempel und einem physikalischen Parameter, vorliegen.

Vis5D bietet die Einbindung von kartographischen und topographischen Modellen, verschiedene Projektionsarten (generisch rektile Projektion, zylindrisch äquidistante Projektion, Lambert-Projektion, stereographische Projektion), Koordinatensysteme (gleichmäßig oder unregelmäßig unterteilt, generische Einheiten oder Kilometer/Grad) sowie die Möglichkeit zur Kompression.

Mit Vis5D können horizontale und vertikale Querschnitte oder Volumenmodelle von Phänomenen, die sich auf einen oder mehrere physikalische Parameter beziehen, in diskreten Zeitschritten oder als Animation betrachtet werden. Solche Sequenzen können auch auf Video festgehalten werden [HP94].

Benutzeranfragen mit dem Ziel der Visualisierung eines Ausschnitts aus der Datenmenge werden an das Datenbanksystem gestellt. Sie enthalten Angaben zu den fünf Dimensionen und zu den Visualisierungsmodi.

Wie unscharf diese Angaben sein dürfen, hängt vom Stand der Implementierung der Metadaten ab. Mit Hilfe dieser Metadaten wird eine Anfrage in einer internen Repräsentation erzeugt.

Das Ziel dieser internen Anfrage ist, die Adressen der auszulesenden netCDF-Raumobjekte im selben Anwendungsprogramm zusammen mit Angaben zu Dimensionen und Visualisierungsmodi einer Konvertierungsroutine zu übergeben.

Die Übergabe von Daten an ein Visualisierungssystem geschieht durch Auswahl von Dateien, Auslesen bestimmter Datenbereiche und anschließender Konvertierung in ein vom Visualisierungssystem lesbares Format.

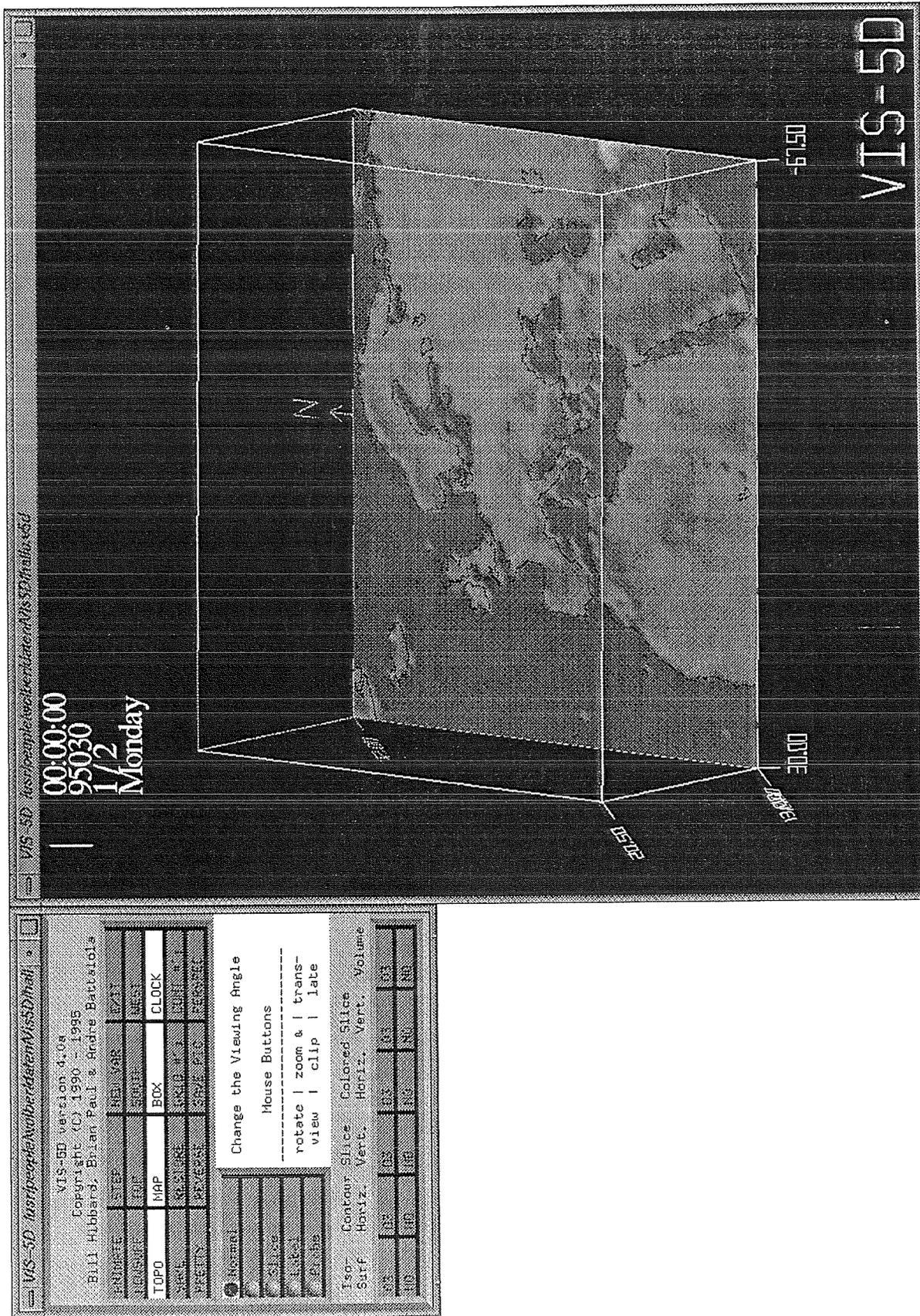
Die Konvertierungsroutine benutzt Schnittstellenfunktionen sowohl von Vis5D als auch von netCDF. Ihre Aufgabe ist zum einen die Initialisierung des Visualisierungssystems. Dazu gehören die Auswahl von Projektion, Kompression und Koordinatensystemen.

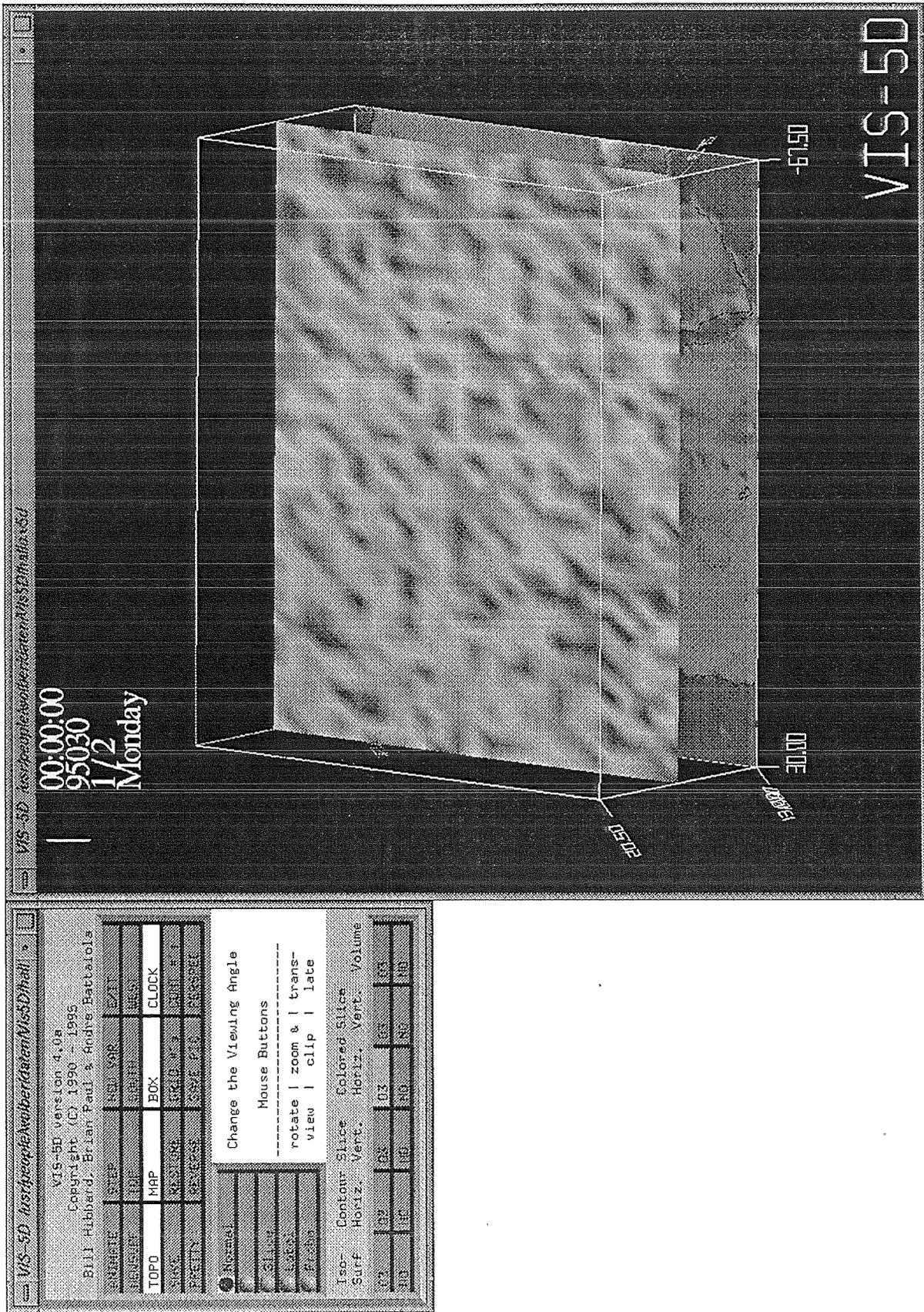
Zum anderen liest sie aus jeder Eingabedatei den zum gewünschten Raumausschnitt gehörenden Datenbereich aus. Zusammen mit Zeitstempel und Parameter werden diese Daten in ein Vis5D-spezifisches Format überführt. Die erzeugte Datei dient schließlich als Eingabedatei für einen Aufruf des Visualisierungssystems.

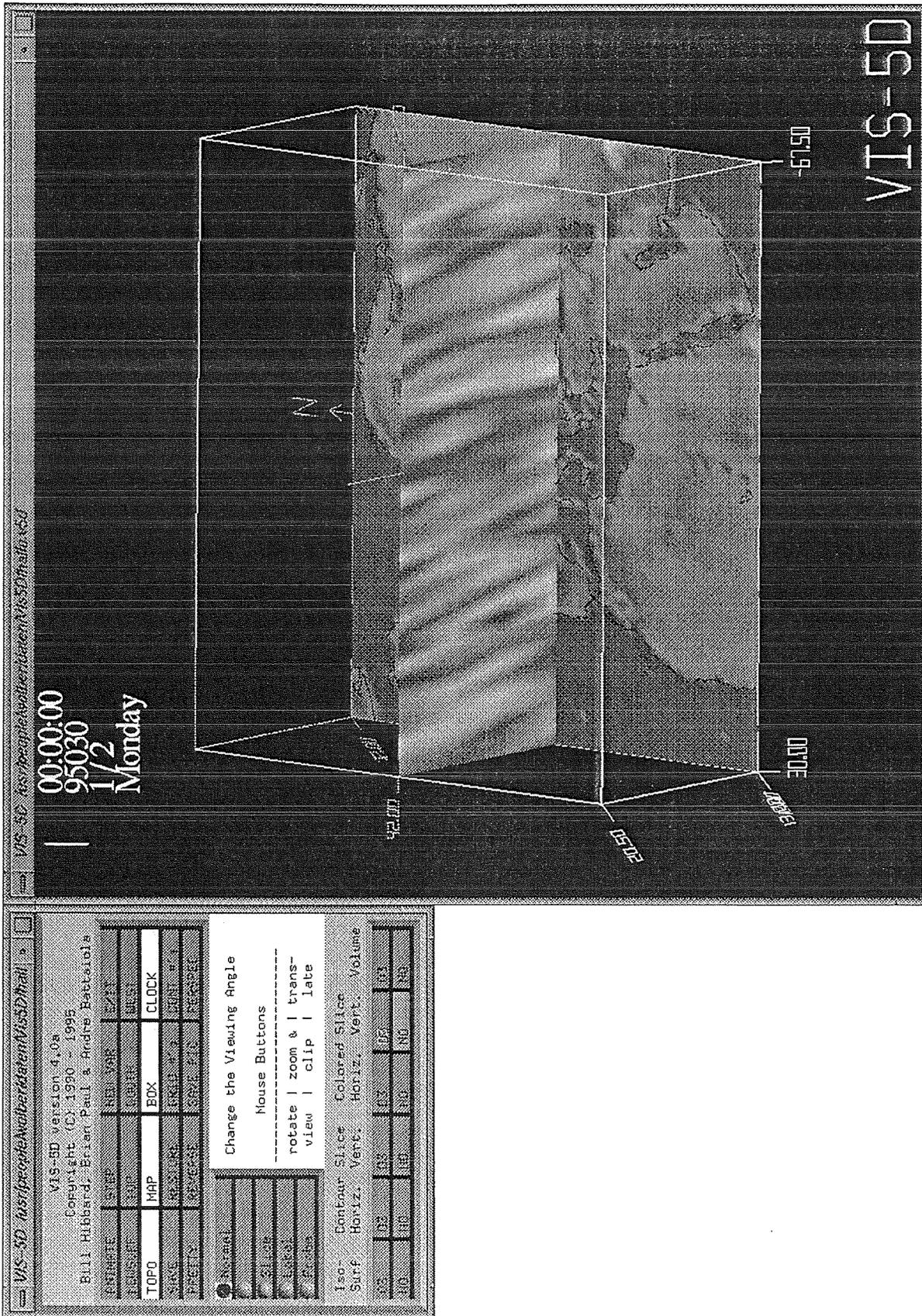
Die folgenden Abbildungen zeigen die Visualisierung des Ergebnisses einer Anfrage an die räumliche Datenbank bezüglich der Konzentration von O<sub>3</sub> und NO in dem durch die Karte dargestellten Raumausschnitt zu zwei Zeitpunkten.

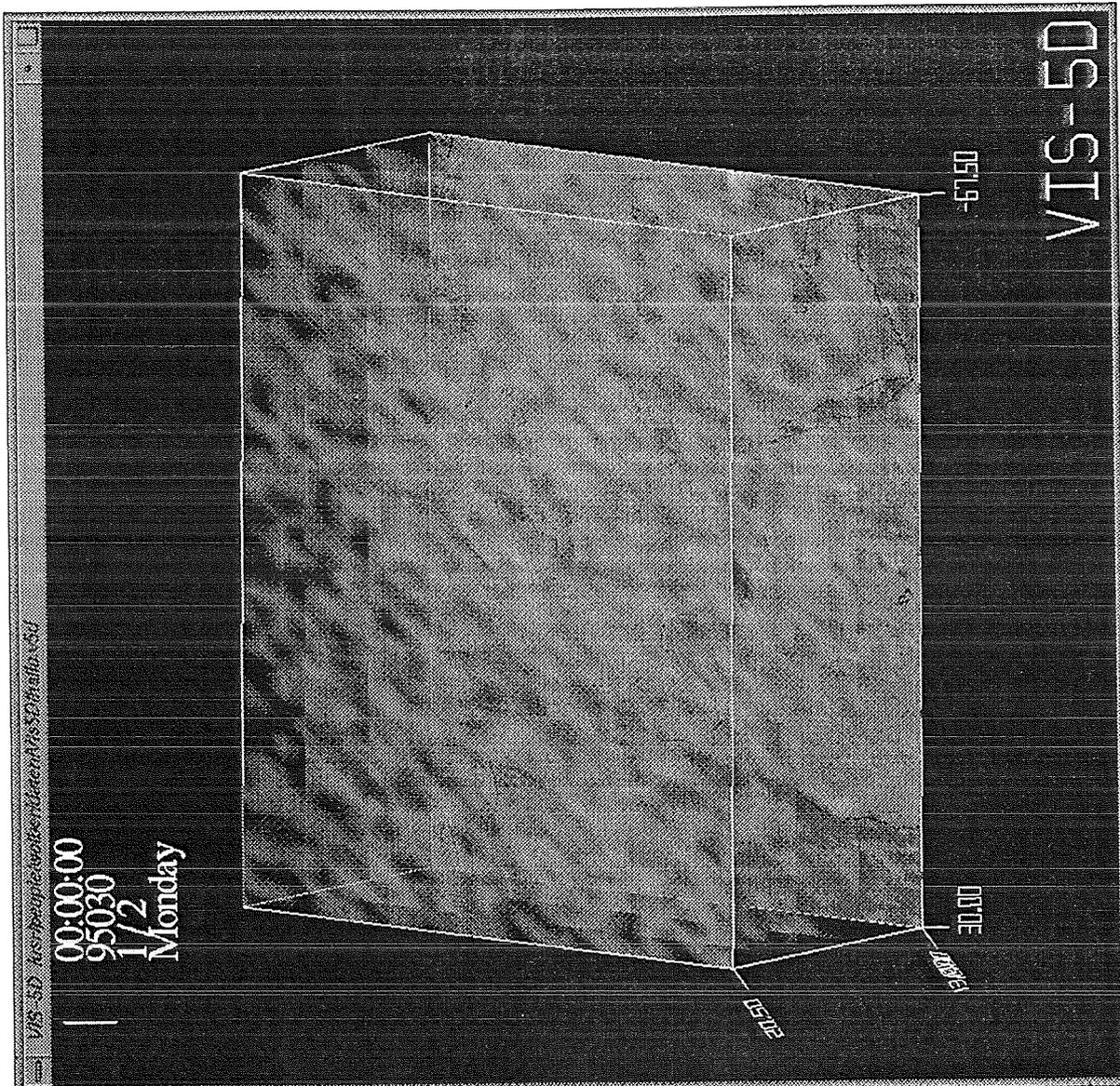
Auf dem ersten Bild ist neben dem Vis5D-Control-Panel, das die interaktive Visualisierung der Daten erlaubt, ein Vis5D-Fenster mit einem Zeit/Datum-Stempel und einer Box mit dem Raumausschnitt in rektilinear Projektion zu sehen.

Den weiteren Abbildungen liegen wegen fehlender realer Meßdaten Zufallszahlen als Eingabedaten zugrunde. Es werden ein horizontaler und ein vertikaler Querschnitt sowie ein Volumenmodell gezeigt.









**VIS-5D** *kurzprogramm zur Visualisierung von 5D-Datensätzen*

VIS-5D version 4.0a  
 Copyright (C) 1990 - 1995  
 Billi Hibbard, Brian Paul & Andre Battiola

BYRDATE	STEP	RED	GRN	BLU
UNSORT	TOP	SHR	SHR	SHR
TOPO	MAP	BOX	CLOCK	
SURF	RESUME	SLICE	VERT	SLICE
PRESN	REVERSE	SHR	GRN	BLU

Resend

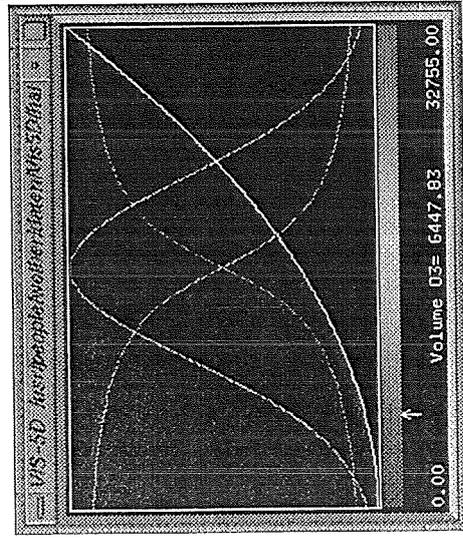
**Change the Viewing Angle**

Mouse Buttons

rotate | zoom & | trans-  
 view | clip | late

---  
 Isc Contour Slice Colored Slice  
 Surf Horiz. Vert. Horiz. Vert. Volume

03	03	03	03	03	03
03	03	03	03	03	03



## 6.3 Bewertung

Drei Punkte sind für die Bewertung der hybriden Lösung von Bedeutung:

- Die Vereinigung der Vorteile von Datenbanksystemen mit den Vorteilen von wissenschaftlichen Datenformaten bei der Modellierung einer Datenstruktur für das konkrete Problem.
- Die Umgehung des Impedence Mismatch bei der Integration von Informationssystemen mit Visualisierungssystemen.
- Die Sonderstellung der Dimension Zeit.

Eine hybride Realisierung bietet sehr gute Möglichkeiten zur Datenmodellierung und zur Realisierung einer fünfdimensionalen Zugriffsstruktur.

Der effiziente Zugriff auf Dateien wird durch ein OODBMS gewährleistet, der effiziente Zugriff innerhalb dieser Dateien auf die zu visualisierenden Datenbereiche durch ein wissenschaftliches Datenformat. Das OODBMS stellt Transaktionsmechanismen für die Dateimanipulation zur Verfügung.

Die Abarbeitung deklarativer Benutzeranfragen und die Auswahl von Dateien zur Übergabe an ein Visualisierungssystem werden durch das OODBMS übernommen, ebenso die Modellierung und Verwaltung von Metadaten der durch das wissenschaftliche Datenformat erstellten Dateien.

Die Vorteile einer hybriden Lösung sind insbesondere in der einfachen Integration der räumlichen Datenbank mit einem Visualisierungssystem zu sehen. Erstens haben viele Visualisierungssysteme Schnittstellen zu wissenschaftlichen Datenformaten.

Zweitens bietet die Lösung trotz des fünfdimensionalen Datenzugriffs die Übergabe von Dateien mit dreidimensionalen Daten ans Visualisierungssystem, was wegen der Einschränkung vieler Visualisierungssysteme bezüglich der Eingabedaten sehr wichtig ist.

Daneben erlauben wissenschaftliche Datenformate eine explizite Zerteilung des Datenraums in direkt ans Visualisierungssystem zu übergebende Dateien. Verteilte, heterogene Umgebungen sind bei einer hybriden Lösung genauso wenig problematisch wie der Austausch von Daten mit anderen Wissenschaftlern.

Die Konzepte für die Realisierung des konzeptuellen Modells mit einem Implementierungs- oder physikalischen Modell beruhen auf einer Verallgemeinerung der Realisierung zwei- oder dreidimensionaler räumlicher Datenstrukturen auf fünf Dimensionen.

Dagegen ergibt sich bei einer hybriden Lösung eine Sonderstellung für die Zeit. Dies eröffnet die Perspektive, daß die räumlichen Dimensionen aus der Dimension Zeit abgeleitet werden können, falls die SDF-Dateien nicht den gesamten betrachteten Raumausschnitt überdecken sollten.

Insgesamt läßt sich feststellen, daß die Abarbeitung der zu erwartenden Anfragen beim interaktiven Arbeiten mit der Visualisierungskomponente mit geringen Zugriffszeiten auf die Datenmenge verbunden ist.

Die Datenstruktur wurde so entworfen, daß bei der Anfragebearbeitung nur auf wenige Dateien zugegriffen werden muß. Nur bei besonderen Anwendungen (Aufnahme von Videos, ...) ist mit einer höheren Zugriffszeit zu rechnen, da große Datenmengen adressiert, ausgelesen und übergeben werden müssen.

## 7 Ausblick

Ziel dieser Arbeit war die Gestaltung der Integration einer Visualisierungskomponente in ein wissenschaftliches Informationssystem zur Atmosphärenforschung.

Hierfür wurde eine räumliche Datenbank unter Betrachtung der Schnittstellen zum Gesamtsystem und zum Visualisierungssystem entworfen. Die Realisierbarkeit dieser Lösung wurde durch die Entwicklung eines Prototyps erprobt.

Die hybride Realisierung bietet einerseits einen informationsorientierten Datenzugriff mit einer fünfdimensionalen Zugriffsstruktur und andererseits die Zerteilung des Datenraums in explizit adressierbare Dateien, denen eine dreidimensionale Datenstruktur zugrunde liegt.

Dadurch ist eine effiziente Abarbeitung von Benutzeranfragen und eine informationsorientierte Aktivierung verschiedener Visualisierungssysteme ebenso möglich wie die Verwendung heterogener Hardwareplattformen und der Austausch von Daten mit anderen Wissenschaftlern.

Eine weitere zu lösende technische Aufgabe ist die reale Implementierung eines leistungsfähigen Visualisierungs-Verwaltungs-Systems.

Zunächst einmal ist ein sorgfältiger Softwareentwurf erforderlich. Für den objektorientierten Teil der räumlichen Datenbank muß entschieden werden, welche Funktionalitäten als Anwendungsprogramm und welche als Methoden realisiert werden sollen.

Daneben muß ein leistungsfähiges Konzept für die Übergabe von Daten und Programmaufrufen zwischen objektorientierter Datenbank, wissenschaftlichem Datenformat und Visualisierungssystem erstellt werden.

Außerdem zu betrachten ist der Aspekt der Benutzungsschnittstelle. Hier ist nach einer Lösung zu suchen, die natürlichsprachige oder graphische Eingaben ermöglicht.

Inhaltliches Ziel der weiteren Arbeit ist die Ausgestaltung der Visualisierungskomponente zu einem Werkzeug für die Wissensakquisition, also für die Ableitung von Wissen bezüglich der Interaktion verschiedener Spurengase in einem vierdimensionalen Raum (eine zeitliche und drei räumliche Dimensionen).

Das Visualisierungs-Verwaltungs-System sollte so ausgebaut werden, daß die Benutzer dabei unterstützt werden, das durch Beobachtungen mit Hilfe der Visualisierungskomponente erlangte Wissen zu beschreiben und zu halten.

Grundsätzlich gilt, daß Visualisierungstechniken ein sehr leistungsfähiges Hilfsmittel für die übersichtliche Darstellung großer Datenmengen und komplizierter Sachzusammenhänge sind. Zu entwickeln ist ein Mechanismus für die Generierung wissenschaftlicher Hypothesen mit Hilfe der Visualisierungskomponente.

Konkret sind also zwei Probleme zu lösen: Erstens müssen geeignete Visualisierungsmethoden zur Ableitung von Wissen aus den Beobachtungsdaten erarbeitet werden. Zweitens sind Methoden zur Formalisierung dieses aus der Visualisierungskomponente abgeleiteten Wissens zu entwickeln.

Bisher wurde lediglich die technische Seite der Integration einer Visualisierungskomponente betrachtet. Die inhaltliche Seite, also die Frage, wie die Wissensableitung selbst erfolgen soll, ist noch weitgehend ungeklärt.

## A Abkürzungsverzeichnis

API	Application Programming Interface
DBS	DataBase System
DBMS	DataBase Management System
GIS	Geographical Information System
HDF	Hierarchical Data Format
KBMS	Knowledge Based Management System
netCDF	NETwork Common Data Format
OODBMS	ObjectOriented DataBase Management System
OpDBS	Operational DataBase System
SDB	Spatial DataBase
SDF	Scientific Data Format
VMS	Visualization Management System
VS	Visualization System

## Literatur

- [A<sup>+</sup>92] D. Abel et al., editors. *5th International Symposium on Spatial Data Handling*, Charleston, South Carolina, 1992.
- [AO93] D. Abel and Beng Chin Ooi, editors. *Advances in spatial databases*, Singapur, 1993. Springer-Verlag.
- [Arm88] M. Armstrong. Temporality in spatial databases. In *Proceedings on LIS/GIS*, pages 880–889, 1988.
- [ATB90] K.K. Al-Taha and T. Barrera. Temporal data and GIS. In *Proceedings on LIS/GIS*, pages 244–254, 1990.
- [B<sup>+</sup>89] A. Buchmann et al., editors. *Design and Implementation of Large Spatial Databases*, Santa Barbara, California, 1989. Springer-Verlag.
- [B<sup>+</sup>90] S. M. Bell et al. Handling four dimensional geo-coded data. In *International symposium on spatial data handling*, pages 918–927, 1990.
- [B<sup>+</sup>93] R.D. Bergeron et al. Developing a data model. In *Database Issues for Data Visualization*, pages 3–15. Springer-Verlag, 1993.
- [BF91] R. Bill and D. Fritsch. *Grundlagen der Geo-Informationssysteme*. Wichmann, 1991.
- [Bil93] R. Bill. Raum und Zeit - neue Herausforderungen an Geo-Informationssysteme aus dem Umweltbereich. In O. Günther, K.P. Schulze, and J. Seggelke, editors, *Umweltanwendungen graphischer Informationssysteme*, pages 255–264. Wichmann, 1993.
- [BK<sup>+</sup>91] A. Beller, T. Kittel, et al. A temporal GIS prototype for global change research. In *Proceedings on LIS/GIS*, Atlanta, 1991.
- [DW89] B. E. Davis and R. Williams. The five dimensions of GIS. In *Proceedings on LIS/GIS*, pages 50–58, 1989.
- [EGH<sup>+</sup>93] J. Ebbinghaus, O. Günther, G. Hess, J. Lamperts, and W.-F. Riekert. Erfahrungen mit objektorientierten Datenbanksystemen für Geo- und Umweltanwendungen im Projekt GODOT. FAW Ulm, 1993.
- [EK94] J. L. Encarnacao and V. Kühn. Visualisierungstechniken in Umweltanwendungen. *it+ti*, 4/5, 1994.

- [Enq95] Enquete-Kommission 'Schutz der Erdatmosphäre' des 12. Deutschen Bundestages, editor. *Mehr Zukunft für die Erde*. Economica Verlag Bonn, 1995.
- [FCF92] A. Frank, I. Campari, and U. Formentini, editors. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. Springer-Verlag, 1992.
- [FO90] H. Fischer and F.-S. Olesen. Fernerkundung des Zustands und der Zusammensetzung der Atmosphäre. In *Umweltforschung Umwelttechnik*. KfK, 1990.
- [G+93] G. Grinstein et al. System integration issues. In *Database Issues for Data Visualization*, pages 16–24, 1993.
- [Heu92] A. Heuer. *Objektorientierte Datenbanken*. Addison-Wesley, 1992.
- [HLW91] N.W.J. Hazelton, F.J. Leahy, and I.P. Williamson. On the design of temporally-referenced, 3-d geographical information systems: development of four-dimensional GIS. In *Proceedings on LIS/GIS*, pages 357–372, Atlanta, 1991.
- [HMN93] H. Hagen, H. Müller, and G. Nielson, editors. *Focus on Scientific Visualization*. Springer-Verlag, 1993.
- [HP94] W. Hibbard and B. Paul. *Vis5D Version 4.0*. Space Science and Engineering Center, University of Wisconsin - Madison, 1994.
- [Kap94a] E. Kapetanios. Anforderungen an die Datenbanktechnologie zur Verwaltung von Satellitendaten für die Fernerkundung atmosphärischer Parameter. In *Umweltdatenbanken*, pages 87–109, 1994.
- [Kap94b] E. Kapetanios. Design specification of the scientific database system for MIPAS satellite experiment. KfK 5277, 1994.
- [Kap95] E. Kapetanios. A scientific knowledge base for extracting and justifying scientific hypotheses in atmospheric research. In *2nd International Conference on Building and Sharing of Very Large-Scale Knowledge Bases*, pages 132–142, Eschede, 1995.
- [KBS93] D. T. Kao, R. D. Bergeron, and T. M. Sparr. An extended schema model for scientific data. In *Database Issues for Data Visualization*. Springer-Verlag, 1993.
- [KK95] E. Kapetanios and R. Kramer. A knowledge-based system approach for scientific data analysis and the notion of metadata. In *2nd IEEE International Symposium on Mass Storage Systems*, 1995.

- [Koc93] P. Kochevar. Database management for data visualization. In *Database Issues for Data Visualization*. Springer-Verlag, 1993.
- [Kra94] H. Kramer. *Observation of the Earth and its Environment*. Springer-Verlag, 1994.
- [Lee93] J. P. Lee. Data exploration interactions and the exbase system. In *Database Issues for Scientific Visualization*, pages 118–137, 1993.
- [LL94] S. Lang and P.C. Lockemann. Datenbankeinsatz. Skriptum zur Vorlesung, Universität Karlsruhe, 1994.
- [LMK94] P.C. Lockemann, H. Müller, and C. Kilger. Implementierung von Datenbanksystemen. Skriptum zur Vorlesung, Universität Karlsruhe, 1994.
- [LS89] P.C. Lockemann and J.W. Schmidt. *Datenbank-Handbuch*. Springer-Verlag, 1989.
- [LT92] R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*. Academic Press, 1992.
- [Moe94] G. Moerkotte. Transaktionsverwaltung. Skriptum zur Vorlesung, Universität Karlsruhe, 1994.
- [Nat93] National Center for Supercomputing Applications. *HDF Reference Manual*, 1993.
- [NB93] G. M. Nielson and D. Bergeron, editors. *Visualization '93*. IEEE Computer Society Press, 1993.
- [Nie89] J. Nievergelt. 7+-2 criteria for assessing and comparing spatial data structures. In *Design and Implementation of Large Spatial Databases*, pages 3–27, 1989.
- [Obj94] Object Design, Burlington. *ObjectStore User Guide*, February 1994. Release 3.0 for System V.4.
- [RDE93] R. Rew, G. Davis, and S. Emmerson. *NetCDF User's Guide*. Unidata Program Center, 1993.
- [Sam89] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1989.
- [TCS+93] A. Tansel, J. Clifford, R. Snodgrass, et al. *Temporal Databases*. The Benjamin/Cummings Publishing Company, 1993.
- [War91] G. Warnecke. *Meteorologie und Umwelt*. Springer-Verlag, 1991.