

# Technisches Handbuch



## Semantic Data Dictionary SDD

Der „Information-Broker“ für das  
Spatial Data Warehouse

Version 1.0.5 Professional Edition

Autor: Markus Widmer  
ESRI Geoinformatik AG  
Beckenhofstrasse 72  
CH-8006 Zürich

Datum: 18.08.16 11:56

Datei: SDD Techn. Handbuch V1.0.5.doc

Version: 1.0.5



# Inhaltsverzeichnis

<b>1</b>	<b>EINLEITUNG.....</b>	<b>6</b>
1.1	WAS SIND DIE AUFGABEN EINES „INFORMATION BROKER“ ? .....	6
1.2	WAS WILL MAN ERREICHEN ? .....	6
1.3	REPORT, AD-HOC-ABFRAGE ODER ANALYSE .....	7
1.4	METADATEN – DER SCHLÜSSEL ZUM ERFOLG .....	8
<b>2</b>	<b>WAS IST DAS SEMANTIC DATA DICTIONARY ? .....</b>	<b>9</b>
2.1	REFERENZSYSTEM FÜR GEODATEN.....	9
2.1.1	<i>Datenmodellierung</i> .....	10
2.1.2	<i>Beschreibung aller Phasen des Vorgehensmodells</i> .....	10
2.2	RECHTEVERGABE- UND VERWALTUNGSSYSTEM.....	10
2.2.1	<i>Anwender- und Applikationsobjekt-Rechte</i> .....	10
2.2.2	<i>Sprachunabhängigkeit</i> .....	11
2.3	REPOSITORY FÜR APPLIKATIONSOBJEKTE .....	11
<b>3</b>	<b>DIE KLASSENKATEGORIEN DES SEMANTIC DATA DICTIONARY .....</b>	<b>12</b>
3.1	ARBEITSSITZUNG.....	12
3.2	ENTITY-RELATIONSHIP-MODELL.....	12
3.2.1	<i>Entität</i> .....	13
3.2.1.1	<i>Entitätgruppe</i> .....	13
3.2.1.2	<i>Member-Entität</i> .....	13
3.2.1.3	<i>Objektlose Entität</i> .....	13
3.2.1.4	<i>Business-Entität</i> .....	13
3.2.2	<i>Beziehung</i> .....	13
3.2.3	<i>Attribut</i> .....	14
3.2.4	<i>Geodaten-Quelle</i> .....	14
3.2.5	<i>Geodaten-Verbindung</i> .....	14
3.3	RECHTE .....	14
3.3.1	<i>Anwender</i> .....	14
3.3.2	<i>Benutzergruppen</i> .....	14
3.3.3	<i>Privilegien</i> .....	14
3.4	SPRACHE .....	15
3.5	ABFRAGEN .....	15
3.5.1	<i>Geometrische Restriktion</i> .....	15
3.5.2	<i>Attributive Restriktion</i> .....	15
3.5.3	<i>Logische Restriktion</i> .....	15
3.6	MENGE .....	16
3.6.1	<i>Objektmenge</i> .....	16
3.6.2	<i>Beziehungsmenge</i> .....	16
3.6.3	<i>Infopaket</i> .....	16
3.6.4	<i>Beziehungs-Infopaket</i> .....	16
3.6.5	<i>Mengenattribut</i> .....	17
3.6.6	<i>Formel</i> .....	17
3.7	ANSICHTEN .....	17



3.7.1	<i>Thema</i> .....	17
3.7.2	<i>Layer</i> .....	17
3.8	DARSTELLUNGSNORMEN.....	17
3.8.1	<i>Style</i> .....	18
3.8.1.1	Punktsymbol.....	18
3.8.1.2	Liniensymbol.....	18
3.8.1.3	Flächensymbol.....	18
3.8.1.4	Beschriftung.....	19
3.8.2	<i>Klassifikation (geplant)</i> .....	19
3.8.3	<i>Qualitative Einzelwertdarstellung (geplant)</i> .....	19
3.8.4	<i>Diagramme (geplant)</i> .....	19
3.8.4.1	Balkendiagramm.....	19
3.8.4.2	Kuchendiagramm.....	20
3.8.4.3	Pfeildiagramm.....	20
<b>4</b>	<b>GEOASSISTENTEN</b> .....	<b>21</b>
4.1	ADMINISTRATIONSASSISTENT.....	22
4.1.1	<i>Pflege des Datenmodells</i> .....	22
4.1.2	<i>Verwaltung der Nutzungsrechte</i> .....	23
4.1.3	<i>Verwaltung der Applikationsobjekte</i> .....	23
4.1.4	<i>Dokumentation des Leistungsprofils</i> .....	23
4.2	FORTSCHREIBUNGSASSISTENT (GEPLANT).....	23
4.3	ANALYSEASSISTENT.....	23
4.3.1	<i>Generische Dialoge für sachbezogene Raumanalysen</i> .....	24
4.3.1.1	Ansicht-Dialog.....	24
4.3.1.2	Abfrage-Dialog.....	24
4.3.1.3	Mengen-Dialog.....	25
4.3.2	<i>Bereitstellung von Informationen</i> .....	25
4.3.3	<i>Multiuser-Betrieb über Semantic Data Dictionary SDD</i> .....	25
4.4	INFOASSISTENT.....	25
4.4.1	<i>Client/Server-Applikation</i> .....	25
4.4.1.1	InfoAssistent InternetMapServer IMS.....	26
4.4.1.2	Statuslosigkeit.....	27
4.4.2	<i>InfoAssistent Browser</i> .....	28
<b>5</b>	<b>SYSTEMARCHITEKTUR</b> .....	<b>29</b>
5.1	MEHRSCICHTIGE CLIENT/SERVER-ARCHITEKTUR.....	29
5.2	VERTEILTE CLIENT/SERVER-ARCHITEKTUR.....	29
5.3	AUFBAU DES SDD-SERVERS.....	31
5.4	BASIS FÜR GEOASSISTENTEN.....	32
<b>6</b>	<b>KOMPONENTEN FÜR DIE ANWENDUNGSPROGRAMMIERUNG</b> .....	<b>33</b>
6.1	MAPMANAGER.....	34
6.2	SETMANAGER.....	35
6.3	QUERYMANAGER.....	36
<b>7</b>	<b>REALISIERUNGSKONZEPT</b> .....	<b>37</b>
7.1	STANDARDS.....	37
7.1.1	<i>Objekt-Orientierung</i> .....	37
7.1.2	<i>Klassen, Klassenbegriff</i> .....	37



7.1.3	<i>Kommunikation</i> .....	37
7.1.4	<i>Datenbank-Zugriff</i> .....	37



# 1 Einleitung

Neben den klassischen GIS-Funktionalitäten „Verwaltung-Erfassung-Analyse-Kartographie“ treten bei grossen Anwendungsumgebungen zunehmend informations-technologische Aspekte in den Vordergrund, wie z.B. die Integration von Informationssystemen, die Zugänglichkeit für einen grossen Nutzerkreis, die langfristige Wartbarkeit, Skalierbarkeit etc. Im Zentrum steht dabei das Ziel, die Investitionen in einer umfassenden und aktuellen Datenbasis im Sinne eines Spatial Data Warehouse für strategische Planungen nutzbar zu machen.

## 1.1 Was sind die Aufgaben eines „Information Broker“ ?

Der „Information Broker“ übernimmt eine zentrale Stellung in der Verteilung von Informationen in einem Unternehmen. Die Aufgaben umfassen: Informationen vorzudefinieren und dann bereitzustellen oder auf Anfrage Informationen auszuwerten. In jedem Fall geht es darum, vielen verschiedenen Anwendern in flexibler Weise den Zugriff auf mehr Informationen zu geben.

Unter dem technischen Aspekt bringt eine solche Öffnung keine fundamentalen Änderungen für die Architektur eines Spatial Data Warehouse mit sich, denn die Benutzerkreise sind nur zusätzliche Clients, die auf den Server zugreifen. Die Änderung umfasst lediglich eine zusätzliche Schicht in der Systemarchitektur, über die aggregierte Informationen bereitgestellt werden.

## 1.2 Was will man erreichen ?

Das Ziel ist eine Abkehr vom Expertentum. Damit sich Spatial Data Warehouses rechnen, muss das in ihnen gespeicherte Wissen dem ganzen Unternehmen zugänglich gemacht werden – nicht nur einzelnen GIS-Spezialisten.

Dass das Wissen einen bedeutenden Faktor für den Erfolg in einem Unternehmen darstellt, ist nicht neu. Aber ausgerechnet in der Informationstechnologie ist die Umsetzung noch annähernd nicht so weit, wie sie sein könnte. Das liegt zu einem guten Teil daran, dass Informationstechnologien vielfach stärker unter dem Aspekt der Technologie als der Informationen gesehen wird. Viele Technologien haben nur dann eine Berechtigung, wenn es ihnen gelingt, das Wissen allen potentiellen Nachfragern im Unternehmen, unter Berücksichtigen von Datenschutzaspekten, in einfacher Weise bereitzustellen. Der Anwender will dabei ohne spezialisierte Kenntnisse der zugrundeliegenden Daten oder einer Abfragesprache zu Informationen gelangen und, wenn möglich in seiner Terminologie sowie in seiner Muttersprache mit dem System kommunizieren.

Ein weiteres wesentliches Ziel ist die Verkürzung, Beschleunigung und Vereinheitlichung der Kommunikationswege in einem Unternehmen. Dabei steht die gemeinsame Nutzung von Geodaten, Sachinformationen und Analyseresultaten unter Berücksichtigung des Datenschutzes im Vordergrund.



### 1.3 Report, Ad-hoc-Abfrage oder Analyse

Business Intelligence Tools helfen, Daten aus dem Spatial Data Warehouse zu Information zu verdichten, die für die Entscheidungsfindung relevant sind.

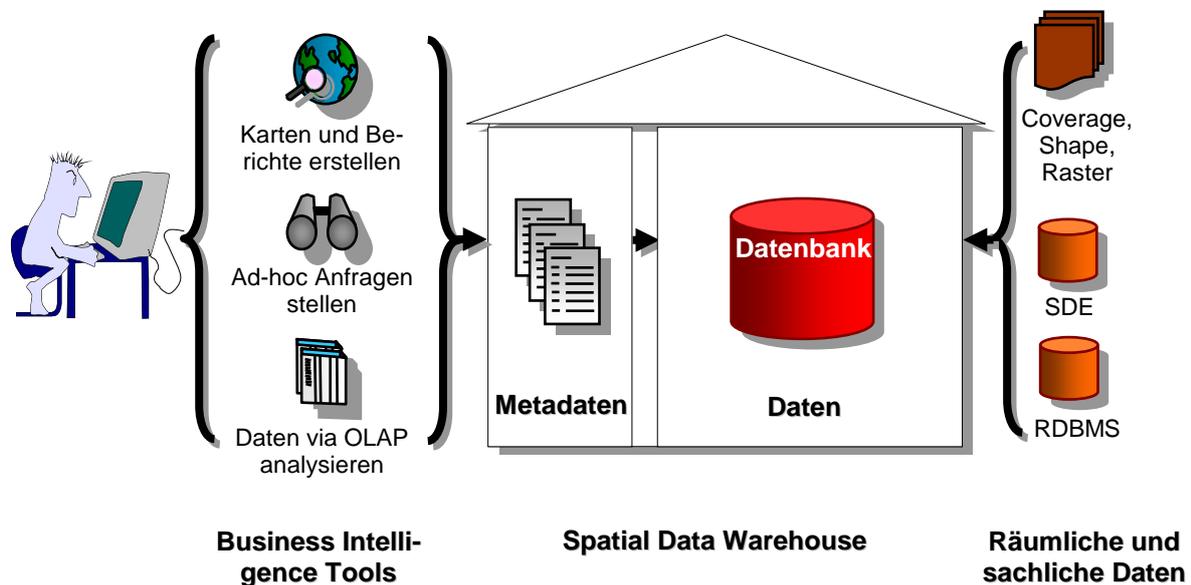


Abb. 1: Entscheidungsfindung

Die sogenannten Business Intelligence Tools lassen sich in unserem Fall in drei verschiedene Kategorien unterteilen:

- **Karten und Berichte**  
Standardberichte oder Karten fassen regelmässig benötigte Informationen in übersichtlicher Form zusammen.
- **Ad-hoc-Anfragen**  
Das sind Anfragen nach spezifischen oder individuellen Informationen, die Standardberichte in dieser Form nicht liefern.
- **Datenanalyse**  
Im Vordergrund steht die Analyse oder Interpretation der Daten. Das bekannteste Datenanalyse-Verfahren ist OLAP (Online Analytical Processing). Analysewerkzeuge bereiten Daten so auf, dass sie in verschiedenen Verdichtungen und Dimensionen betrachtet werden können.

## 1.4 Metadaten – der Schlüssel zum Erfolg

Metadaten sind Informationen über Daten und bilden den Kern des Systems. Das *Semantic Data Dictionary* speichert und verwaltet diese Metadaten. Im wesentlichen ist das die Repräsentation des Applikationsdatenmodells mit seinen Entitäten und deren Beziehungen untereinander (Entity/Relationship-Modell), sowie der graphischen Repräsentation dieser Entitäten in der Karte – dem Darstellungsmodell.

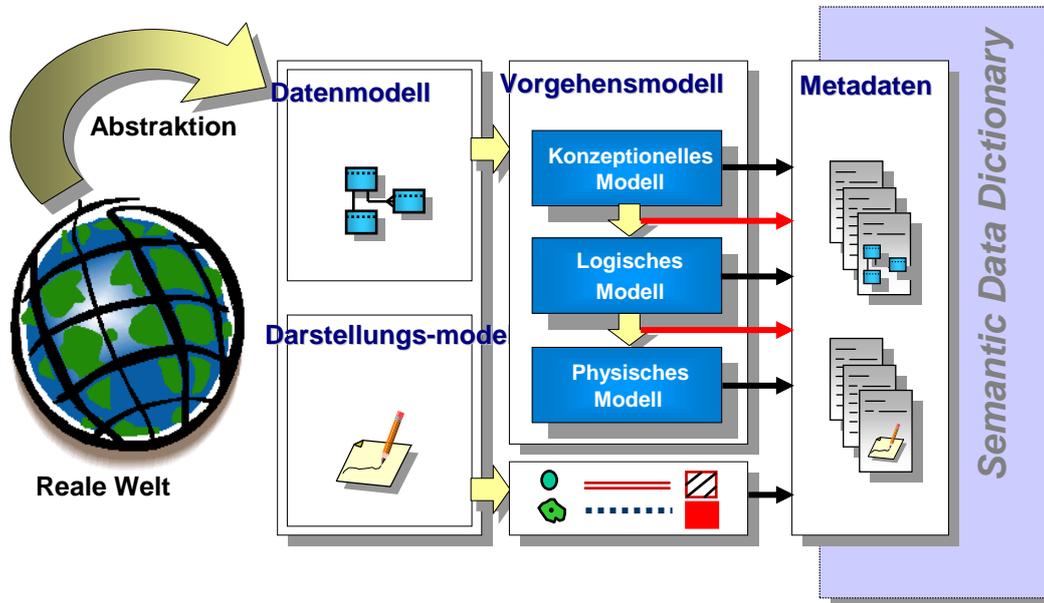


Abb. 2: Das SDD beschreibt Informationen über Geodaten

Die nachfolgenden häufigsten Fragen, die ein Anwender an Metadaten hat, können vom *Semantic Data Dictionary* beantwortet werden.

- Welche Daten sind vorhanden?
- Wo sind diese Daten erhältlich?
- Wie kann darauf zugegriffen werden?
- Welche Einschränkungen gibt es?
- Was bedeuten die Dateninhalte?

Im weiteren wird das SDD ergänzt durch eine Verwaltung von Anwendern, Benutzergruppen und deren Rechten, sowie durch die Speicherung und durchgängige Verwendbarkeit von anwenderdefinierten Kartensichten, Objektmengen und Abfragedefinitionen.

## 2 Was ist das Semantic Data Dictionary ?

Unter dem Begriff Semantic Data Dictionary SDD werden verschiedene zentrale Dienste eines Raumbezugssystems RBS zusammengefasst, die nicht nur die Geodatenhaltung selbst betreffen, sondern Verwaltungsdienste um die Nutzung der Geodaten.

- **Referenzsystem für Geodaten**

Der Grundstein für die flexible Gestaltung einer Applikation ist die Beschreibung aller (Geo-) Daten im System. Diese Metadaten bilden den Kern des SDD und umfassen im wesentlichen Entitäten, die zugehörigen Attribute und die Beziehungen (*Entity-/Relationship-Modell*).

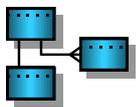
- **Rechtevergabe- und Verwaltungssystem**

Der Datenschutzaspekt spielt bei einer gemeinsamen Nutzung von Daten eine wichtige Rolle. Dieser Anforderung wird das SDD gerecht, indem jeder Zugriff koordiniert und kontrolliert erfolgt. Berechtigungen von Nutzern und Sicherheiten von Applikationsobjekten gegenüber den Nutzern werden definiert und im SDD verwaltet.

- **Repository für Applikationsobjekte**

Das SDD betreibt ein Repository für Applikationsobjekte. In dieser Eigenschaft ist das SDD ein zentraler Behälter bzw. Server für benutzerdefinierte Analysen (Abfragen), permanente Analyseresultate (Mengen), benutzerdefinierte Sichten (Ansichten) und Kartenlayouts (Darstellungsnormen) von Geodaten.

### 2.1 Referenzsystem für Geodaten



Das SDD, als Referenzsystem für Geodaten, enthält eine Beschreibung aller im System abgebildeten Klassen von Realweltobjekten sowie alle systemrelevanten Eigenschaften. Im wesentlichen sind das Entitäten, die zugehörigen Attribute und die Beziehungen. Ausserdem wird die Art und Weise der Abbildung dieser Objekte durch das System beschrieben.

- *Metadaten*

Metadaten sind Daten über Daten. Sie beschreiben die Inhalte der Geodaten.

- *Data Dictionary*

Beschreibt die Struktur der Geodaten.

- *Semantisches Modell*

Beschreibt die anwendungsbezogene Strukturierung von Geodaten.



### 2.1.1 Datenmodellierung

Beim Design eines Informationssystems nimmt die Datenmodellierung eine zentrale Stellung ein. Im Verlauf der Datenmodellierung wird im wesentlichen festgelegt, welche Aspekte der Realwelt in welcher Art und Weise im System abgebildet werden.

In der Regel wird folgendermassen vorgegangen:

1. *Erstellen eines konzeptionellen, vom Zielsystem unabhängigen Datenmodells*

Dabei wird in enger Zusammenarbeit mit den Fachanwendern definiert, welche Objekte für das System relevant sind, mit Hilfe welcher Attribute sie beschrieben werden können und welche Beziehungen zwischen den Objekten existieren. Das dabei entstehende Modell wird in einer leicht verständlichen Weise, meist in Form eines Entity-Relationship-Modell (E/R-Modell), beschrieben.

2. *Erstellen eines logischen, systemspezifischen Datenmodells*

In dieser Phase wird das Modell des vorhergehenden Schritts auf die Konzepte des Zielsystems abgebildet. Es wird z.B. festgelegt, wie eine Beziehung zwischen zwei Entitäten im Zielsystem abgebildet wird. Dabei gibt es meistens verschiedene Möglichkeiten diese Abbildung zu realisieren und bedingt sehr gute Kenntnisse des Zielsystems.

3. *Erstellen des physischen Datenmodells*

In dieser Phase wird das Modell physisch implementiert.

### 2.1.2 Beschreibung aller Phasen des Vorgehensmodells

Im SDD wird eine detaillierte Beschreibung des konzeptionellen, des logischen und des physischen Datenmodells - inklusive Informationen über die Abbildung der verschiedenen Modelle - abgelegt.

Durch die Beschreibung aller im System abgebildeten Geodaten wird es möglich, mit kleinem Aufwand sehr flexible, benutzerorientierte Anwendungen zu entwickeln. Sobald alle Funktionen konsequent auf dem SDD basieren, muss bei einer Änderung des semantischen Datenmodells lediglich das SDD nachgeführt werden. Eine Modifikation der Funktionalität ist nicht nötig. Die Funktionalität kann somit in unterschiedlichen thematischen Zusammenhängen ohne weiteren Programmieraufwand verwendet werden. Zudem kann die Interaktion mit dem Benutzer in der ihm vertrauten Begrifflichkeit gestaltet werden.

## 2.2 Rechtevergabe- und Verwaltungssystem

### 2.2.1 Anwender- und Applikationsobjekt-Rechte



Das Raumbezugsystems RBS implementiert eine Rechtevergabe. Nutzer eines RBS haben Nutzer-Rechte und unterliegen den Restriktionen hinsichtlich dieser Rechte. Weiter sind ebenfalls alle Applikationsobjekte Gegenstände dieser Rechte, d.h. der Rechtevergabe und der Zugriffskontrolle unterworfen.

Das SDD in der Eigenschaft als Rechtevergabesystem verwaltet:

- *Anwender und Benutzergruppen*

Es können beliebige Anwender zu beliebig vielen Benutzergruppen - und umgekehrt - definiert werden.

- *Anwender-Rechte*

Berechtigung des Nutzers bezüglich eines Applikationsobjektes. D.h. darf der Anwender ein Applikationsobjekt erzeugen, lesen, löschen oder die Rechte ändern.

- *Applikationsobjekt-Rechte*

Sicherheit eines Applikationsobjekts gegenüber dem Nutzer. D.h. Definition von Privilegien (kein, lesen, schreiben) bezüglich des Anwenders.

Das Szenario der Authentifizierung ist ein Vergleich des Anwender-Rechts (der Berechtigung) mit dem Applikationsobjekt-Recht (der Sicherheit). Herrscht Übereinstimmung, ist der Zugriff legal und zugelassen. Bei Nichtübereinstimmung ist der Zugriff illegal und wird zurückgewiesen.

## 2.2.2 Sprachunabhängigkeit



Das SDD ist multilingual. D.h. die Anforderung der Sprachunabhängigkeit wird unterstützt, indem alle Beschreibungen im SDD in verschiedenen Sprachen abgelegt und verwaltet werden können. Somit kann jeder Anwender in seiner Sprache mit dem System kommunizieren.

## 2.3 Repository für Applikationsobjekte

Das SDD betreibt ein Repository für Applikationsobjekte. Applikationsobjekte sind neben den Entitäten, den Beziehungen, den Sachdaten, der Sprache, den Anwendern, den Benutzergruppen und deren Rechte:



- *Abfragen*

Benutzerdefinierte Analysen auf die Geodaten



- *Mengen*

Permanente Analyseresultate



- *Ansichten*

Sicht auf die Geodaten / Inhalt einer Karte



- *Darstellungsnormen*

Benutzerdefinierte Darstellung von Geodaten / Symbolisierung der Karte

### 3 Die Klassenkategorien des Semantic Data Dictionary

Der Kern des Semantic Data Dictionary SDD ist die formale Beschreibung aller Geodaten im System, unter Berücksichtigung der Datenschutzaspekte. Die nachfolgende Graphik zeigt, welche Klassenkategorien im SDD vorhanden sind und wie diese zusammen spielen.

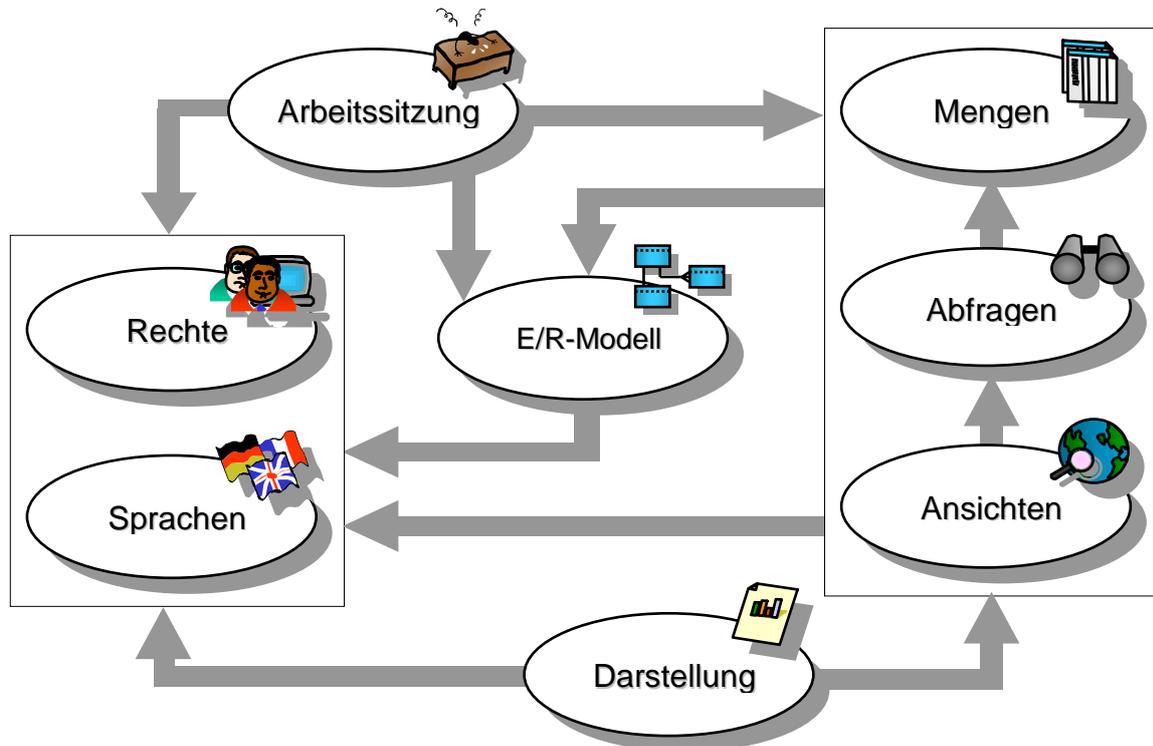


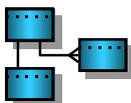
Abb. 3: Klassenkategorien des Semantic Data Dictionary

#### 3.1 Arbeitssitzung



Jeder GeoAssistent (Client) muss sich beim SDD (Server) mit Benutzername und Passwort anmelden und wird von der Rechteverwaltung des SDD registriert und kontrolliert. Ist die Anmeldung erfolgreich, wird dem GeoAssistent ein Interface für die Applikationsobjekte zur Verfügung gestellt.

#### 3.2 Entity-Relationship-Modell



Das SDD enthält eine Beschreibung aller im System abgebildeten Klassen von Realweltobjekten sowie alle systemrelevanten Eigenschaften. Ausserdem wird die Art und Weise der Abbildung dieser Objekte durch das System beschrieben.

### 3.2.1 Entität

Eine Entität ist ein Objekt der realen oder der gedachten Welt. Sie wird definiert durch einen Satz von identischen Attributen und Beziehungen zu anderen Entitäten.

#### 3.2.1.1 Entitätsgruppe

Dieser Typ von Entität ist der Normalfall und besitzt eine Beschreibung von zugehörigen Sachdaten und Beziehungen zu anderen Entitäten. Weiter kann dieser Typ in sogenannte Member-Entitäten gegliedert werden. Dieses Konstrukt ist eingeführt worden, um eine Modellierung von z.B. der Gruppe *Strassen* mit den Members *Autobahn*, *Bundesstrasse*, *Feldweg*, ... zu unterstützen.

#### 3.2.1.2 Member-Entität

Neben der Abbildung von Entitätsgruppen wird erlaubt, diese in sogenannten Member-Entitäten zusammenzufassen. Dabei haben alle Entitätsmember identische Sachdaten der zugehörigen Entitätsgruppe und unterscheiden sich lediglich bei der Typisierung

#### 3.2.1.3 Objektlose Entität

Rasterkarten, Orthophotos, etc. werden in Form einer objektlosen Entität im SDD beschrieben und verwaltet. Eine objektlose Entität hat keine Sachdaten und keine Beziehungen zu anderen Entitäten.

#### 3.2.1.4 Business-Entität

Business-Entitäten sind Entitäten ohne direkten geographischen Bezug, wie z.B. ein Eigentümerverzeichnis. Sie haben aber wie die Entitätsgruppe Sachdaten und Beziehungen zu anderen Entitäten.

### 3.2.2 Beziehung

Jede Beziehungen gilt zwischen zwei Entitäten. Dabei können auch mehrere solcher Beziehungen definiert sein. (z.B. zwischen Kante und Knoten können die beiden Beziehungen *'beginnt an'* und *'endet an'* existieren.)

Dabei werden nachfolgende Beziehungstypen unterschieden:

- *Objektreferenz*  
Attribut des referenzierenden Objekts. Enthält den Objektschlüssel (ID) des referenzierten Objekts.
- *Namensreferenz*  
Sprechender Identifikator des referenzierten Objekts. Kann eindeutig aus einem Attribut des referenzierenden Objekts abgeleitet werden.
- *Hierarchischer Schlüssel*  
Sprechender Identifikator des referenzierten Objekts. Kann eindeutig aus einem Attribut des referenzierenden Objekts mittels einem Substring abgeleitet werden.

Für die vorgängig aufgelisteten Beziehung müssen Regeln festgelegt werden, wie der sprechende Identifikator des referenzierten Objekts eindeutig abgeleitet werden kann.

Eine Beziehung besteht aus genau zwei Beziehungshälften. Die Beziehung wird einmal aus Sicht der referenzierenden (1. Beziehungshälfte) und aus Sicht der referenzierten Entität (2. Beziehungshälfte) beschrieben.

Dieser Abstraktionsgrad ist eingeführt worden, um später ev. Mehrstufige-Beziehungen unterstützen zu können.

Es werden nachfolgende Kardinalitäten von Beziehungen unterstützt:

- *1:c* (1:1, bedingt)
- *1:1* (1:1, unbedingt)
- *1:cn* (1:n, bedingt)
- *1:n* (1:n, unbedingt)

### 3.2.3 Attribut

Jede Attributbeschreibung ist genau einer Entität zugeordnet.

Es werden nachfolgende Attributtypen unterstützt:

- *Zahl (Integer-Wert)*
- *Gleitkommazahl (Float-Wert)*
- *Text (String)*
- *Datum (Date)*

### 3.2.4 Geodaten-Quelle

Alle Elemente des Entity-Relationship-Modells müssen physikalisch in einer Datenquelle vorliegen. Das können einzelne Kacheldateien von Rasterkarten, Shape-Dateien oder SDE-Tabellen sein.

### 3.2.5 Geodaten-Verbindung

Die Geodaten-Verbindung ist der Ordner in welchen die Geodatenquellen abgelegt sind. Ist der zugrundeliegend Typ von Entität „Objektlos“ (Rasterkarte) so können verschiedene Geodaten-Verbindungen existieren, ansonsten nur genau eine. Weiter beschreibt die Geodaten-Verbindung lediglich die Datenquelle, wie sie heisst, wo sie liegt und wie darauf zugegriffen werden kann.

## 3.3 Rechte



Alle Applikationsobjekte unterliegen einem koordinierten und kontrollierten Zugriff. Dabei spielt der Datenschutzaspekt eine wichtige Rolle. Jeder Kontakt (Login) zum SDD wird von der Rechteverwaltung registriert und die Berechtigungen eines Nutzers zu den Applikationsobjekten kontrolliert.

### 3.3.1 Anwender

Im SDD werden alle Anwender, die Zugang zum Server haben, verwaltet. Dabei kann ein Anwender zu verschiedenen Benutzergruppen gehören. Bei Neuanlegung eines Applikationsobjektes bekommt der Erzeuger sogenannte 'Verwaltungsrechte' auf das Objekt.

### 3.3.2 Benutzergruppen

Die Berechtigung eines Anwender zu den Applikationsobjekten wird über Benutzergruppen (Role) gesteuert und verwaltet. Jeder Anwender ist mindestens einer Benutzergruppe zugeordnet, nämlich seiner eigenen.

### 3.3.3 Privilegien

Es werden im SDD drei verschiedene Privilegien unterstützt:

- *lesen (read)*
- *schreiben (write)*
- *verwalten (grant)*

Dabei gilt zu beachten, dass Privilegien auf die Container-Klasse und auf das einzelne Applikationsobjekt gesetzt werden können. Ein *Schreibrecht* auf einer Container-Klasse bedeutet, dass Applikationsobjekte dieser Klasse erzeugt bzw. gelöscht werden können. Ein *Verwaltungsrecht* hingegen bedeutet, dass die Privilegien geändert werden können.

### 3.4 Sprache



Die Benutzernamen von Applikationsobjekten (Entitäten, Attributen, Beziehungsworte, Mengen, etc.) können in mehreren Sprachen definiert werden. (siehe 2.2.2)

### 3.5 Abfragen



Abfragen sind benutzerdefinierte Verfahren auf die Geodaten. Dabei bezieht sich das Verfahren auf genau eine Entität. Diese Verfahren werden im SDD gespeichert und verwaltet. Eine Abfrage besteht aus beliebig vielen „Simple-“ oder „Complex-Expressions“, die mit AND, OR oder MINUS miteinander verbunden werden.

Für die Spezifikation von Abfragen werden folgende Restriktionsmöglichkeiten, in beliebiger Kombination, zur Verfügung gestellt:

- Geometrische Restriktion
- Restriktion über Sachdaten
- Selektion aufgrund logischer Beziehungen zwischen Objekten

Abfragen können auf Basis bereits bestehender Mengen formuliert werden oder sich auf den Gesamtdatenbestand beziehen

#### 3.5.1 Geometrische Restriktion

Eine Abfrage kann räumlich eingeschränkt werden. Als räumliche Einschränkungen können reguläre Polygone, irreguläre Polygone und Referenzen auf bestehende Geometrien dienen.

Reguläre Polygone sind:

- Rechteck / Quadrat
- Kreis

Irreguläre Polygone sind:

- Drei- bis beliebige Vielecke
- Korridor

#### 3.5.2 Attributive Restriktion

Eine Abfrage kann attributiv (d.h. Sachdaten-bezogen) eingeschränkt werden. Als Sachdaten-bezogene Einschränkung werden alle Attribute der ausgewählten Datenquelle (Entität/Menge) zugelassen. Alle gängigen Operationen wie  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $=$ , etc. werden unterstützt. Als Vergleichswerte werden Parameter zugelassen, die erst beim Ausführen der Abfrage benötigt werden.

#### 3.5.3 Logische Restriktion

Die logische Einschränkung nimmt eine Sonderstellung ein. Logische Einschränkungen basieren auf modellbedingten Beziehungen (relationships) zwischen Entitäten und ermöglichen das Navigieren



durch das semantische Datenmodell. Ist bei der Abfrage eine logische Restriktion definiert, so beziehen sich die geometrischen und attributiven Restriktionen immer auf die Basis-Entität, das Resultat jedoch auf diejenige Entität, die bei der logischen Restriktion als Ziel-Entität ausgewählt worden ist.

### 3.6 Menge



Die Resultatmengen von Abfragen sind konzeptionell unterschiedslos Mengen, technisch gesehen jeweils Tabellen. Die logische Richtigkeit ist genau beim Zeitpunkt der Erstellung gewährleistet. Es werden nur homogene Mengen unterstützt, d.h. alle Features in der Menge müssen derselben Entität angehören.

Mengen lassen sich anhand ihres Informationsgehalts in unterschiedliche Klassen zuordnen, denen jeweils spezifische Methodenvorräte zu eigen sind. Es werden vier Klassen von Mengen unterschieden:

- *Objektmenge*
- *Beziehungsmenge*
- *Infopaket*
- *Beziehungs-Infopaket*

#### 3.6.1 Objektmenge

Eine Objektmenge ist bildlich eine Tabelle mit beliebig vielen Zeilen und eins bis ca. 250 Spalten, wobei die erste Spalte den Objektschlüssel (Key/ID) beinhaltet. Sind mehrere Spalten vorhanden, so sind zu dieser Objektmenge noch zusätzliche Wertattribute vorhanden.

#### 3.6.2 Beziehungsmenge

In diesem Mengentyp werden Beziehungen zwischen zwei Objekten gespeichert. Es werden nur 1:n - Beziehungen unterstützt. Die Beziehungsinformation kann entweder aus dem SDD oder ad-hoc mittels der GIS-Funktionalität 'Verschneidung' aufgebaut werden. Weiter können zusätzliche Wertattribute gespeichert werden.

#### 3.6.3 Infopaket

Infopakete sind spezielle Mengen. Wegen ihrer Eigenschaft als multidimensionale Attributmengen können Infopakete nicht unmittelbar visualisiert werden. Infopakete erfordern eine zusätzliche analytische Behandlung, bevor ihre Inhalte präsentiert werden können. Daraus ist ersichtlich, dass auf Infopakete stets die Funktion *Sicht bilden (Pivoting)*, die eine Aggregation nach anzugebenden Kriterien darstellt, angewendet wird.

Hierbei sind folgende Schritte möglich:

- *Einschränkung der Entität(en)*
- *Auswahl von Gliederungsattributen (Gliederungsmerkmal, qualitatives Attribut)*
- *Auswahl von Ausprägungen (Feature-IDs, Objektschlüssel)*
- *Auswahl einer statistischen Funktion bei Wertattributen*

Das Resultat von *Sicht bilden* ist jeweils eine Objektmenge mit Wertattributen.

#### 3.6.4 Beziehungs-Infopaket

Ein Beziehungs-Infopaket hat die selben Eigenschaften wie das normale Infopaket, jedoch ein zusätzliches räumliches Merkmal wie die Beziehungsmenge.



### 3.6.5 Mengenattribut

Mengenattribute sind immer Wertattribute (im Falle von Infopaketen Wert- oder Gliederungsattribute), die genau einer Menge zugeordnet sind.

Es werden nachfolgende Mengenattributtypen unterstützt:

- *Zahl (Integer-Wert)*
- *Gleitkommazahl (Float-Wert)*
- *Text (String)*
- *Datum (Date)*

Zusätzlich wird eine Typisierung unterstützt. Eine Typisierung ist im Normalfall ein Gliederungsattribut, wobei bei der Menge lediglich ein Identifikator (Key/ID) gespeichert ist und der Wert selbst in Lookup-Tabellen vorliegt.

### 3.6.6 Formel

Für die arithmetische Berechnung eines neuen Mengenattributes kann eine beliebige Formel mit Variablen definiert werden. Dabei werden die Variablen mit Mengenattributen einer Menge ersetzt.

## 3.7 Ansichten



Ansichten sind der sichtbare Inhalt des Kartenfensters und werden sowohl in thematischer als auch in räumlicher Hinsicht definiert. Es existiert im SDD ein eigenständiger Verwaltungsmechanismus für Ansichten. Eine Ansicht ist baumartig aufgebaut und beinhaltet die Übereinanderlagerung mehrerer *Themen*, die ihrerseits aus mehreren, nicht zwangsläufig aufeinanderfolgenden *Layern* (Geodaten-Ebenen) zusammengesetzt sein können. Die einzelnen 'Knoten' des Ansicht-Baums werden mit Ordnungsnummern versehen, so dass die einzelnen Geodaten-Schichten mit der richtigen Darstellungspriorität (Bildaufbauichtung des Stapels) aufgebaut werden können.

### 3.7.1 Thema

Themen sind die oberste Ordnungsstufe (Gliederungsstufe) von inhaltlich, d.h. thematisch zusammengehörigen Geodaten. Themen dienen der inhaltlichen Zusammenfassung und Ordnung von Geodaten. Ein Thema beinhaltet mindestens wiederum ein Thema oder ein Layer.

### 3.7.2 Layer

Ein Layer ist die graphische Abbildung (Geodaten-Ebene) einer Entität. Dabei werden folgende Geodaten-Schichten unterstützt:

- *Objekt Layer (Vektor-Daten)*
- *Objektloser Layer (Raster-Daten)*

## 3.8 Darstellungsnormen



Die Symbolisierung der Geodaten-Schichten eines Kartenfensters können vom Anwender frei definiert und im SDD verwaltet werden.

Es werden in dieser Version folgende Renderer unterstützt:

- *Einfaches Punkt-, Linien- oder Flächensymbol (Anwendbar auf die ganze Geodaten-Schicht)*
- *Beschriftung*

Geplant sind:

- *Klassifikation*
- *Qualitative Einzelwertdarstellung*
- *Dichteraster*
- *Balkendiagramm*
- *Kuchendiagramm*
- *Pfeildiagramm*

### 3.8.1 Style

Jeder Layer ist standardmässig mit einer eigenen Symbolisierung (*style*) versehen. Dabei wird unterschieden, ob die Geodaten-Schicht punkt-, linien- oder flächenförmig ist.

#### 3.8.1.1 Punktsymbol

Die punktförmige Darstellung von Features ist definiert durch den Markerstyle, die Grösse und die Symbolfarbe.

Es werden vorläufig folgende MarkerStyles unterstützt:

- *Kreis*
- *Quadrat*
- *Dreieck*
- *Kreuz*
- *True Type*

#### 3.8.1.2 Liniensymbol

Die linienförmige Darstellung von Features unterscheidet sich zur punktförmigen Darstellung lediglich im Style.

Es werden vorläufig folgende LinienStyles unterstützt:

- *Solid*
- *Punktiert*
- *Strichiert*
- *Strich-Punktiert*
- *Strich-Punktiert-Punktiert*

#### 3.8.1.3 Flächensymbol

Die flächenförmige Darstellung von Features ist definiert durch die Füllart, die Füllfarbe, die Umrisslinie, deren Umrisslinienfarbe und Grösse.

Es werden vorläufig folgende FillStyles unterstützt:

- *Solid*
- *Transparent*
- *Horizontal Schraffiert*
- *Vertikal Schraffiert*
- *Diagonal (links Oben nach rechts Unten) Schraffiert*
- *Diagonal (links Unten nach rechts Oben) Schraffiert*
- *Gekreuzt Schraffiert*
- *Diagonal gekreuzt Schraffiert*
- *Hellgrau*
- *Grau*
- *Dunkelgrau*



### 3.8.1.4 Beschriftung

Zu jedem Layer kann eine Beschriftung definiert werden. Mögliche Optionen sind:

- *Font Typ*
- *Font Farbe*
- *Font Grösse*
- *Font Rotation*
- *Hintergrund (ja/nein)*
- *Attribute (Welche Werte dargestellt werden)*
- *Rotations-Attribut (Welches Attribut für die Rotation verantwortlich ist)*
- *Level-Attribut (Welches Attribut für den Level verantwortlich ist)*
- *Höhen-Attribut (Welches Attribut für die Texthöhe verantwortlich ist)*
- *Symbol-Attribut (Welches Attribut für die Symbolisierung verantwortlich ist)*
- *Spline*
- *Flip*
- *Dublikate*

### 3.8.2 Klassifikation (geplant)

Das SDD erlaubt eine Beschreibung einer Klassifikations-Darstellung (Choropletenkarte) bezüglich eines kontinuierlichen Attributes (numerisch, Gleitkomma). Dabei wird in der Karte jedes Feature gemäss einem numerischen Attributs symbolisiert.

Mögliche Optionen sind:

- *Attribut (Welches Attribut zu klassifizieren ist)*
- *Anzahl Klassen*
- *Klassengrenzen*
- *Klassensymbol*

### 3.8.3 Qualitative Einzelwertdarstellung (geplant)

Bei der qualitativen Einzelwertdarstellung wird ein diskreter Attributwert symbolisiert. Der Attributwert kann alphanumerisch sein.

Mögliche Optionen sind:

- *Attribut (Welche Werte darzustellen sind)*
- *Rotations-Attribute (Welches Attribut für die Rotation verantwortlich ist)*
- *Skalierungs-Attribute (Welches Attribut für die Skalierung verantwortlich ist)*
- *Anzahl Werte*
- *Standard-Symbol (Punkt-, Linien- oder Flächensymbol)*
- *Symbol (Punkt-, Linien- oder Flächensymbol)*

### 3.8.4 Diagramme (geplant)

Am Bezugspunkt der Features können Diagramme gezeichnet werden. Dabei muss im SDD mindestens ein numerisches Attribut definiert sein.

#### 3.8.4.1 Balkendiagramm

Die Flächensymbole je Balken (bzw. definiertem numerischen Attribut) werden mit Farbe und Standard-Grösse im SDD gespeichert und verwaltet. Die Anzahl der Balken richtet sich nach den im SDD definierten Attributen für die Diagramm-Darstellung.

#### 3.8.4.2 Kuchendiagramm

Die Flächensymbole je Sektor (bzw. definiertem numerischen Attribut) werden mit Farbe und Standard-Grösse im SDD gespeichert und verwaltet. Die Anzahl der Sektoren richtet sich nach den im SDD definierten Attributen für die Diagramm-Darstellung.

#### 3.8.4.3 Pfeildiagramm

Eine spezielle Art ist das Pfeildiagramm. Pfeildiagramme dienen der Visualisierung von Beziehungsmengen. Beziehungsmengen legen die Beziehung zwischen zwei Entitäten fest. Jedes Feature einer Entität wird über seinen Labelpunkt mit dem Labelpunkt des zugeordneten Feature der zweiten Entität, in der Beziehungsmenge, mit einem gerichteten Pfeil verbunden. Der Labelpunkt gilt für flächenförmige Entitäten. Bei linienförmigen Entitäten wird der auf der Linie liegende Mittelpunkt der Linienstrecke als Labelpunkt verwendet. Bei punktförmigen Entitäten wird der Punkt selbst als Labelpunkt verwendet.



## 4 GeoAssistenten

Die GeoAssistenten sind in ihrer Funktionalität auf die spezifischen Aufgaben des jeweiligen Anwenders zugeschnitten. Der Funktionsumfang der GeoAssistenten ist so abgestuft, dass einerseits die Spezialisten, andererseits jedoch die vielen sporadischen Anwender ohne GIS-Kenntnis intuitiven Zugriff auf die Datenbasis haben, welche die Grundlage für die Entscheidung bildet.

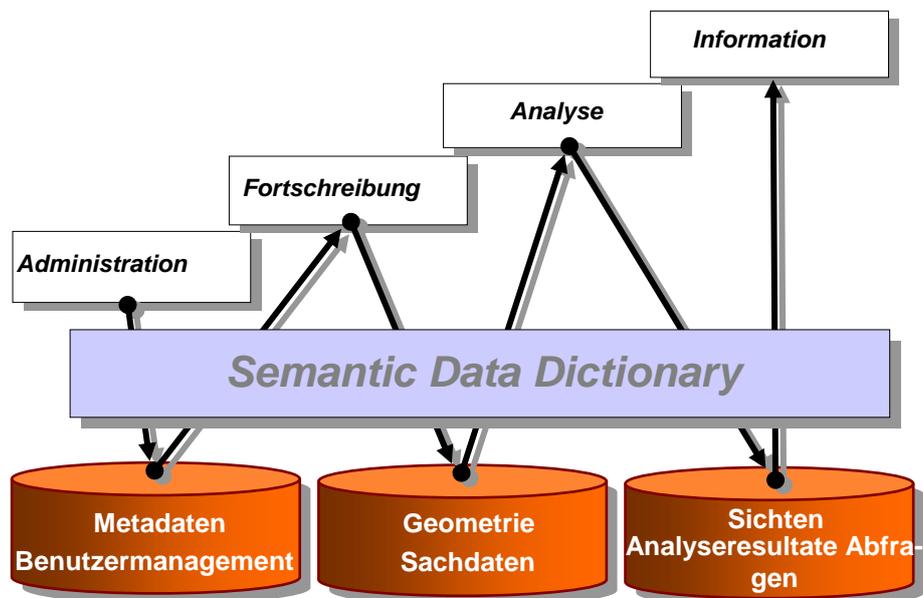


Abb. 4: Wertschöpfungskette

Diese GeoAssistenten (Anwendungen) wurden in Zusammenarbeit mit der Stadt Köln realisiert und nach einer Wertschöpfungskette konzipiert. Dabei wird der gesamte Informationsproduktions-Prozess von der Datenmodellierung über den Aufbau von Geodaten und deren Analyse bis zur Bereitstellung im Internet abgedeckt.

## 4.1 AdministrationsAssistent

Der AdministrationsAssistent ist für die Pflege des Datenmodells und für die Nutzerverwaltung zuständig. Hier werden die Grundstrukturen des Gesamtsystems gepflegt und der Zugriff auf die Daten überwacht. Der AdministrationsAssistent ist das erste Glied der Wertschöpfungskette und das exklusive Werkzeug für den SDD-Administrator.

### 4.1.1 Pflege des Datenmodells

Im SDD können beliebige Geo-Datenmodelle, wie z.B. für Vermessung, Statistik, Umwelt, Kanal, etc. aufgebaut und gepflegt werden. Die dazu benötigten Beschreibungen aller im System abgebildeten Klassen der jeweiligen Realweltobjekte sowie alle systemrelevanten Eigenschaften (*konzeptionelles Modell*) sind im SDD definiert. Im wesentlichen sind das Entitäten, die zugehörigen Attribute und deren Beziehungen (*Entity-/Relationship-Modell*). Ausserdem wird die Art und Weise der Abbildung dieser Objekte durch das System (*logisches Modell*) bis hin zur physischen Implementierung (*physisches Modell*) beschrieben.

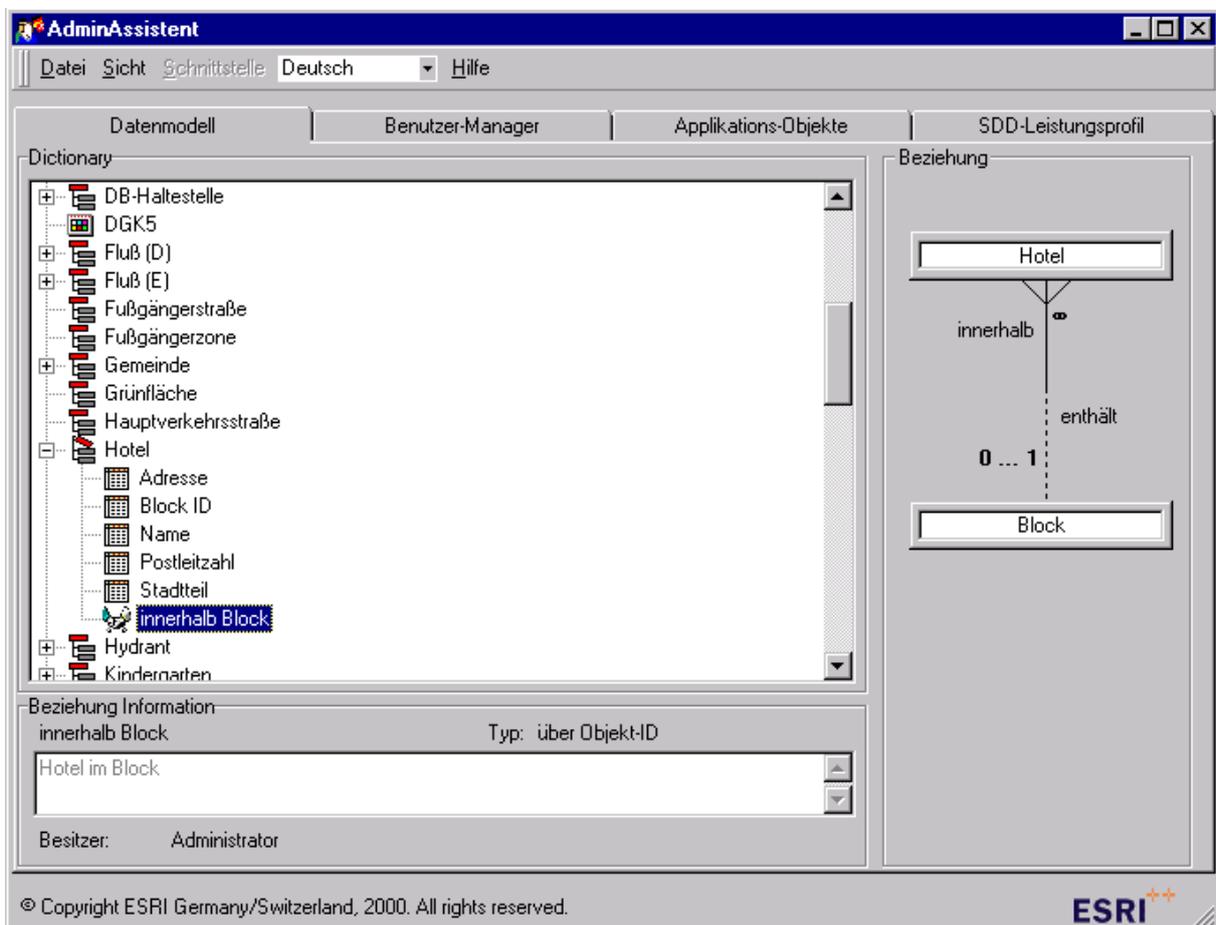


Abb. 5: AdministrationsAssistent

Beim AdministrationsAssistenten werden die oben beschriebenen Schritte der Datenmodellierung nachempfunden. Entitäten und Attribute können definiert, mehrsprachig benannt und beschrieben werden. Neben den beschreibenden Daten können alle systemrelevanten Eigenschaften (Geo- und Sachdatenquellen) eingegeben werden. In vergleichbarer Weise können Beziehungen zwischen den vorhandenen Entitäten hergestellt und die systemspezifische Abbildung auf das jeweilige Zielsystem definiert werden.

### 4.1.2 Verwaltung der Nutzungsrechte

Das SDD kann von den unterschiedlichsten Benutzern verwendet werden. Ein Benutzer wird von der Vielfalt der Daten erschlagen, ein anderer darf aus Datenschutzaspekten auf bestimmte Daten nicht zugreifen und wiederum andere dürfen Daten ändern oder erzeugen.

Um allen Fällen gerecht zu werden sind im SDD sowohl Zugriffsrechte auf Applikationsobjekte für die Benutzer, als auch Sicherheit von Applikationsobjekten vor den Benutzern realisiert. Dazu wird jeder Benutzer einer oder mehreren Benutzergruppen, d.h. Gruppen von gleichermassen berechtigten Benutzern, zugeordnet. Den Benutzergruppen wiederum werden lesende, schreibende oder verwaltende Rechte für die Applikationsklassen (Benutzerrechte) und die Applikationsobjekte (Objektrechte) zugeteilt. Das Zugriffsrecht auf ein bestimmtes Objekt wird schliesslich durch den Vergleich von Benutzerrechten und Objektrechten ermittelt.

### 4.1.3 Verwaltung der Applikationsobjekte

Das SDD betreibt ein Repository für Applikationsobjekte, d.h. definierte Sichten, Layer, Mengen, Abfragen usw. werden im SDD abgelegt und müssen verwaltet werden. Neben der Erstellung neuer Applikationsobjekte können vorhandene Objekte analysiert und modifiziert sowie überflüssig gewordene Objekte entfernt werden. Durch Auflistung von referenzierten Objekten werden Abhängigkeiten unter den Applikationsobjekten transparent gemacht.

In einer Baumstruktur werden neben den Applikationsobjekten auch die Benutzergruppen mit den jeweiligen Nutzungsrechten aufgelistet. Diese können geändert, entzogen oder neu erteilt werden. Eine Gruppierungsfunktion bietet drei verschiedene Sichtweisen der Nutzungsrechte:

- *aus der Sicht der Objekte:*  
welche Benutzergruppe darf was (lesen, schreiben, verwalten) mit mir machen?
- *aus der Sicht der Benutzergruppe:*  
auf welche Applikationsobjekte darf ich wie zugreifen?
- *aus der Sicht der Benutzer:*  
welche maximalen Zugriffsrechte (als Mitglied verschiedener Benutzergruppen) habe ich auf die verschiedenen Applikationsobjekte?

### 4.1.4 Dokumentation des Leistungsprofils

Schliesslich werden vom AdministrationsAssistent Profiling- und Reporting-Funktionen bereitgestellt, mit deren Hilfe man eine schnelle Übersicht über alle im SDD abgelegten Applikationsobjekte erlangen kann.

## 4.2 FortschreibungsAssistent (geplant)

Für den Aufbau der Geodaten und für die konsistente Fortschreibung dient der FortschreibungsAssistent als interaktive Desktop-Applikation.

Weiter ist für den FortschreibungsAssistenten von grosser Bedeutung, Qualitätsziele räumlicher Datenoperationen zu definieren und auch bei umfangreichen Datenänderungen einhalten zu können (Digitalisierungsvorschrift, Fortschreibungsvorschrift). Hierfür ist es notwendig, ein Regelwerk für Objektschlüssel, Geometrien, Attribute und Beziehungen zu erstellen sowie dessen Anwendung konsistent zu erzwingen. Dieses Regelwerk erstreckt sich deshalb auf alle Methoden, die auf einzelne Geoobjekte bzw. Objektmengen angewendet werden.

## 4.3 AnalyseAssistent

Der AnalyseAssistent ist das Werkzeug zur Auswertung und Darstellung raum- und sachbezogener Daten. Dabei werden die zwei Welten GIS und Data Warehouse verknüpft und deren Daten gemeinsam genutzt und analysiert. Der AnalyseAssistent ist das Werkzeug für den Power-User.



### 4.3.1 Generische Dialoge für sachbezogene Raumanalysen

Der AnalyseAssistent ist die zentrale Anwendung für die fachbezogene Erzeugung von Applikationsobjekten. Im wesentlichen sind das thematische Ansichten (Karten), kombinierte räumlich-attributive Abfragen und Analyseresultate (Mengen). Diese Analyseresultate repräsentieren eine bestimmte Teilmenge von Geo-Objekten, die ggf. durch Sachdaten aus dem Data Warehouse näher beschrieben sind. Für den Umgang mit diesen Applikationsobjekten verfügt der AnalyseAssistent über entsprechende Dialoge.

#### 4.3.1.1 Ansicht-Dialog

Der AnalyseAssistent bietet mit dem Ansicht-Dialog interaktiv ausgelegte Tools für die Auswahl und die Navigation durch die räumlichen Daten des Untersuchungsraumes an. Ausgehend von einer Ansicht in einem Untersuchungsgebiet können weitere Sichten und Szenen des Kartenfensters definiert und verwaltet werden. Ansichten werden sowohl in thematischer als auch in räumlicher Hinsicht definiert. Eine Ansicht ist baumartig aufgebaut und beinhaltet die Übereinanderlagerung mehrerer Themen, die ihrerseits aus mehreren Themen und Layern (Geodaten-Ebenen) zusammengesetzt sein können.

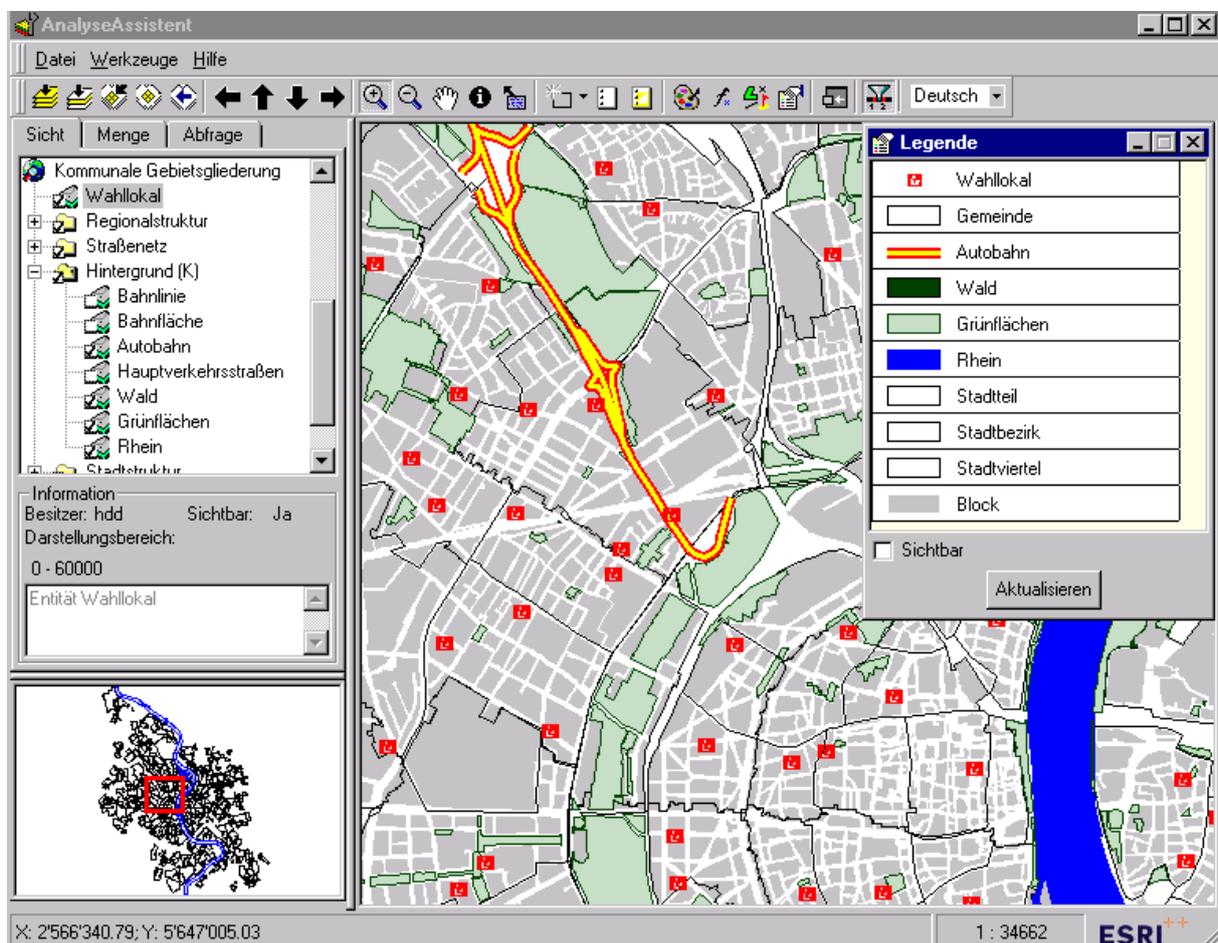


Abb. 6: AnalyseAssistent

#### 4.3.1.2 Abfrage-Dialog

Im Abfrage-Dialog werden benutzerdefinierte Verfahren auf die Geodaten erzeugt und angewendet. Für die Spezifikation von Abfragen werden räumliche, sachbezogene und logische Restriktionsmöglichkeiten in beliebiger Kombination zur Verfügung gestellt. Abfragen können auf der Basis bereits bestehender Analyseresultate (Mengen) formuliert werden oder sich auf den Gesamtdatenbestand beziehen.

### 4.3.1.3 Mengen-Dialog

Das Resultat einer Abfrage ist eine Menge. Dieses Analyseresultat repräsentiert eine bestimmte Teilmenge von Geo-Objekten, die ggf. durch Sachdaten aus dem Data Warehouse näher beschrieben sind. Über Beziehungsmengen, die u.a. durch geometrische Überlagerung erzeugt werden, lassen sich auf den Sachdaten raumbezogene Aggregationen durchführen. Multidimensionale Attributmengen, die als Infopakete oder OLAP-Würfel bezeichnet werden, sind ebenfalls Mengen und können vom AnalyseAssistent bearbeitet werden. Eine typische Operation auf ein Infopaket ist „Sicht bilden“ (Pivoting). Dabei können beliebige Schnitte durch das Infopaket gelegt werden („Anteile der Nationalitäten in der Altersklasse 20-30 Jahre je Stadtteil?“). Damit wird eine Vielzahl von sachbezogenen Sichten auf einen einzigen Datenbestand ermöglicht.

### 4.3.2 Bereitstellung von Informationen

Der AnalyseAssistent wird in erster Linie für die Produktion von neuen Informationen benötigt. Über den Download von raum- und sachbezogenen Analyseresultaten nach ArcView wird das breite Spektrum an spezialisierter Funktionalität dieses Produktes und seiner Erweiterungen erschlossen. Darüberhinaus ist der AnalyseAssistent der Broker für die Bereitstellung von Informationen im Intranet/Internet. Neben einer Vorratshaltung von Informationen kann man über den AnalyseAssistenten auch ad hoc auf unterschiedlichste Informationswünsche reagieren und es können entsprechend Informationen bereitgestellt werden.

### 4.3.3 Multiuser-Betrieb über Semantic Data Dictionary SDD

Ansichten, Abfragen und Mengen werden im serverseitigen Semantic Data Dictionary SDD verwaltet und je nach Rechtevergabe für alle oder ausgewählte Anwender bereitgestellt und freigegeben. Dabei wird nach dem Prinzip „Fat Server, thin Client“ vorgegangen, wobei der Server mehrere Klienten gleichzeitig bedienen kann. Die Ausstattung des AnalyseAssistenten zur Verwaltung der Applikationsobjekte entspricht gängigen Standards moderner GIS-Implementierungen und ermöglicht ein hohes Mass an intuitiver, interaktiver Arbeitsweise.

## 4.4 InfoAssistent

Der InfoAssistent bietet für den grössten und allgemeinsten Kreis von Anwender den informierenden Zugriff auf die Geodaten, sowie auf die aus diesem Datenpool nach fachlichen Kriterien gebildeten Applikationsobjekte. Dies sind primär *Ansichten*, *Abfragen* und *Mengen*, die von den AnalyseAssistenten generiert und als eine allgemeine Ressource im SDD bereitgestellt werden. Die InfoAssistenten bieten die Möglichkeit, Übersichten zu den verfügbaren Geoinformation zu liefern, beliebige Applikationsobjekte bzw. Objektmengen zu verorten und die auf sie bezogenen Sachinformationen abzufragen und darzustellen. Durch den Zugriff auf bereits durch die AnalyseAssistenten vorbereitete Abfragen bzw. Mengen werden Ergebnisse der AnalyseAssistenten einem sehr grossen, nahezu unbegrenzten Anwenderkreis erschlossen. Durch ihre moderne Implementierung in der Form von Internet/Intranet-basierten Browser-Applikationen stellen die InfoAssistenten einen der innovativsten Teile des Gesamtsystems dar. Darüberhinaus erlaubt eine Browser-gestützte Applikation die Nutzung unterschiedlicher Webserver und bietet so den Anwendern des InfoAssistenten die Möglichkeit, beliebige Informationen, also auch Raumbezugsinformationen der unterschiedlichsten Fachabteilungen miteinander in Bezug zu bringen und dadurch den Nutzen der unterschiedlich spezialisierten Modellsichten auf die realweltlichen Gegebenheiten zu mehren.

### 4.4.1 Client/Server-Applikation

Der InfoAssistent besitzt als Web-basierte Client/Server-Applikation zwei primäre Komponenten: einen Web-Browser (HTML, und/oder Java-Applet), und eine Server-Komponente, die als universeller Applikationsserver (InfoAssistent InternetMapServer IMS) im Web-Kontext fungiert.

Der InfoAssistent IMS kann je nach den quantitativen Anforderungen an die Gesamtlösung in einem nahezu beliebigen Ausmass skaliert werden. Anstatt eines einzelnen InfoAssistent IMS-Servers kann

ein Array aus mehreren InfoAssistent IMS-Instanzen auf einer oder mehreren CPUs gestartet werden. Hierbei können thematische und räumliche Differenzierungen der Geodateninhalte auf die jeweiligen Instanzen des InfoAssistent IMS-Karten abgebildet werden (qualitative Skalierung), oder auch eine rein quantitative Skalierung durch Clonierung von InfoAssistent IMS-Instanzen

#### 4.4.1.1 *InfoAssistent InternetMapServer IMS*

Der Applikationsserver ist für den Betrieb mit einem ASP-fähigen (Active Server Pages) Internet-Informationserver von Microsoft konzipiert. Die Aufgabe der multithreaded HTML-Generierung übernimmt dabei die ASP-Scripting-Engine. Innerhalb der ASP-Scripte wird die Möglichkeit genutzt, COM/DCOM-Komponenten aufzurufen und zu nutzen.

Ein SDDMapper-Prozess übernimmt dabei die Aufgabe, multithreaded COM-Zugriffsobjekte im ASP-Script zur Verfügung zu stellen. Dies erfolgt speziell durch einen Teil des Mappers, der ASP-Schnittstellenklasse ("SDDMapper").

Diese Schnittstellenklasse enthält auch das API für die im ASP-Script zur Verfügung gestellten Funktionen. Ein wesentliches Merkmal der Funktionen des APIs ist der Gesichtspunkt der Statuslosigkeit, unter dem sie erstellt worden sind. Dies bedeutet, dass Ergebnisse als Ergebnis und nicht als Zustand im Funktionsserver gehalten werden, sondern die Ergebnisse der Funktionen für nachfolgende Funktionen herangezogen werden.

Der Mapper-Prozess enthält weiters eine Schnittstellenklasse, von der je eine Instanz des Applikationsservers ein Zugriffsobjekt erzeugt. Die Verweise auf die Zugriffsobjekte werden in einem weiteren COM-Prozess ("Master") in einer Mastertabelle abgelegt und als "verfügbar" registriert. Der Master-Prozess hat alle Eigenschaften, um eine optimale Lastverteilung und Applikationsserver-Instanzenverwaltung zu gewährleisten.

Fordert nun eine ASP-Schnittstellenklasse eine bestimmte GIS-Funktion an, so wird kurzfristig aus der Mastertabelle die nächste frei verfügbare Applikationsserverinstanz bzw. dessen Zugriffsobjekt ermittelt, verbunden und als "nicht mehr verfügbar" registriert.

Der ASP-Funktionsaufruf wird so über 2 Threads/ 2 Zugriffsobjekte des Mappers an den Applikationsserver weitergeleitet, das Resultat der Funktion wird in das ASP-Script zurückgeführt.

Der ASP-Funktionsaufruf ist im InfoAssistent IMS als Event realisiert. Dieser Event wird unmittelbar nach Eintritt in den Applikationsserver in ein sogenanntes „Job-Objekt“ verpackt. Das Job-Objekt trägt als Eigenschaften im wesentlichen die Parametrisierung der Funktionsaufrufe aus dem ASP-Script. Das Job-Objekt wird über eine Aktionskette innerhalb des Applikationsservers weitergereicht. Die Aktionskette befragt das Job-Objekt bezüglich dieser Eigenschaften und löst entsprechende Folgeaktionen aus.

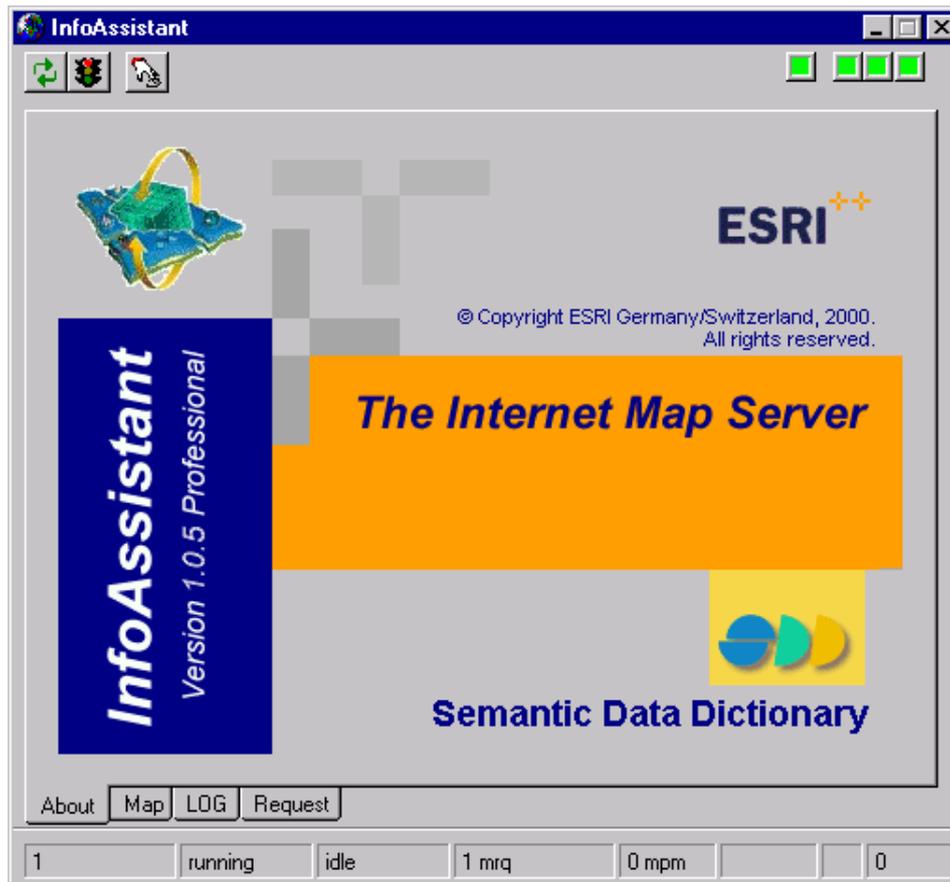


Abb. 6: InfoAssistent InternetMapServer IMS

#### 4.4.1.2 Statuslosigkeit

Die Kommunikation über HTTP ist verbindungslos und bewahrt keinerlei Status. Die Kommunikation erfolgt in einzelnen, voneinander unabhängigen HTTP-Transaktionen. Jede HTTP-Transaktion besteht aus einer vom Browser generierten Funktions-Anforderung, die von einer Server-Antwort bedient wird. Die Server-Antwort versetzt den Browser jeweils in einen neuen *Status*. Dieser Status umfasst alle für die Kartensicht bestimmenden Parameter wie z.B. die Eckkoordinaten, die Sichtbarkeit der Layer (=Themen) und die aktuell angeforderte Funktion (z.B. Zoom In, Zoom Out, Pan, Identify usw.).

Die Architektur des InfoAssistent IMS reflektiert dieses Kommunikationsmodell. Eine Instanz von InfoAssistent IMS bewahrt primär keine Informationen über beantwortete Funktionsrequests. Für eine gewisse Anzahl und eine gewisse Klasse von GIS-Operationen ist dieses Kommunikationsmodell hinreichend. Damit vor dem Hintergrund dieses Kommunikationsmodells eine Kontext-geführte Navigation auf den Kartendokumenten am clientseitigen Browser möglich wird, ist es erforderlich, die auf eine Kartensicht (Kartenszene) bezogene Statusinformation zusammen mit der Kartensicht auf den Browser zu übertragen. Die nächstfolgende Benutzeraktion am Browser basiert auf dieser Kartensicht (Statusinformation). Der durch eine Benutzeraktion erneut ausgelöste Funktionsrequest überträgt die Statusinformation der zugrundeliegenden Kartensicht als erweiterte Parametrisierung, so dass der Server die Beantwortung des Funktionsrequests auf dieser Information (=Status) aufsetzen kann. Auf diese Weise lässt sich eine konsistente GIS-Navigation und damit eine räumlich geleitete Benutzerführung erreichen, der Anwender bewegt sich von einer GIS-Szene zur nächsten.

#### 4.4.2 InfoAssistent Browser

Der InfoAssistent ist ein Web-Browser und als Anwendung für den „Common User“ gedacht. Realisiert als Java-Applet ermöglicht es dem Anwender den grösstmöglichen Komfort. Im InfoAssistent können alle gängigen Kartenfunktionen, wie Zoom, Pan, Identify, etc., ausgeführt werden. Weiter kann auf Informationen zugegriffen werden, die der AnalyseAssistent zur Verfügung gestellt hat. Dies sind primär *Ansichten, Abfragen und Mengen*.

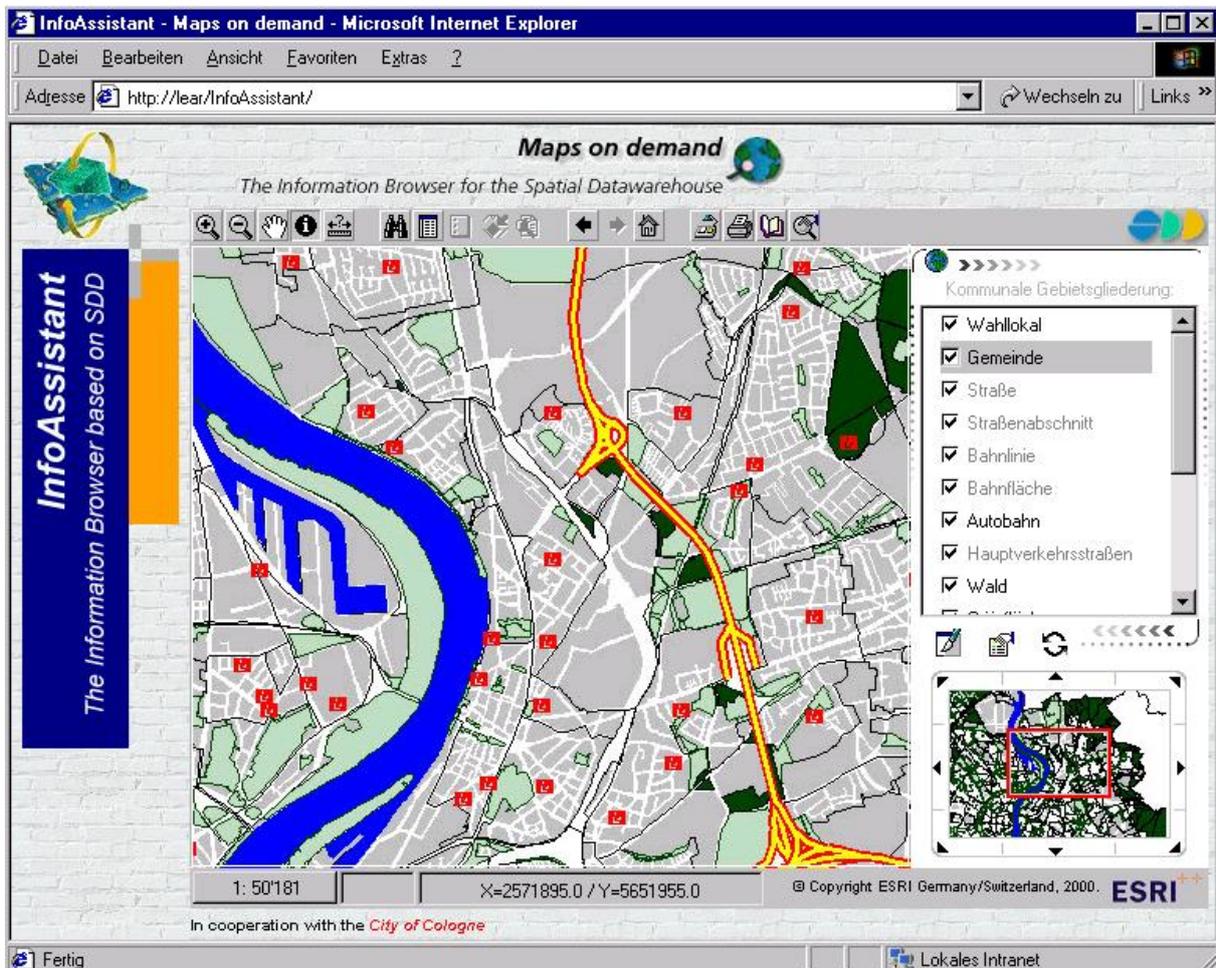


Abb. 7: InfoAssistent

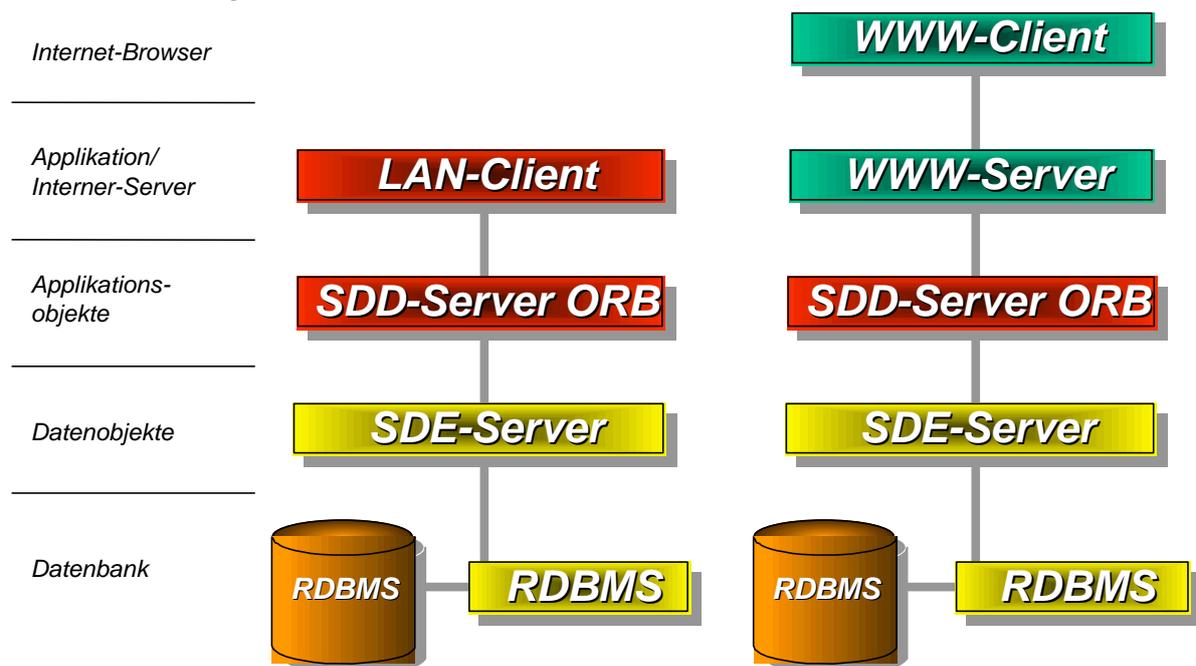
## 5 Systemarchitektur

### 5.1 Mehrschichtige Client/Server-Architektur

In der Informatik spricht man heutzutage von einer mehrschichtigen (n-tier) Client/Server-Architektur. Die erste Schicht bildet der *Data Service*, die zweite der *Application Service* und am Ende das *User Interface*.

- *Data Service*  
Die Basis bildet das Relationale Datenbank Management System RDBMS. Die Datenbank speichert und verwaltet alle Daten. Aufbauend auf der Datenbank stellt die Spatial Database Engine SDE die Datenobjekte bereit.
- *Application Service*  
Die Zwischenschicht bildet das SDD. Das SDD übernimmt die Aufgaben eines Object Request Brokers ORB und stellt Applikationsobjekte zur Verfügung.
- *User Interface*  
Die oberste Schicht ist die Anwendung mit ihrer Benutzeroberfläche.

Abb. 8: Mehrschichtige Architektur



Wird das Intranet/Internet in das System eingebunden, so bildet der World Wide Web WWW - Server eine eigene Schicht. Die oberste Schicht ist in diesem Fall der Browser, der die Schnittstelle zum Anwender ist.

### 5.2 Verteilte Client/Server-Architektur

Aufbauend auf dem SDD-Server können beliebig viele Clients mit unterschiedlichsten fachlichen Anwendungen angeschlossen werden. Informationen, wo sich die Geodaten befinden, liefert der SDD-Server. Wichtig ist, dass im SDD lediglich die Geodaten mit den dazugehörigen Sachinformationen

beschrieben werden und somit die Datenherrschaft unverändert bleibt. Zur standardisierten Bereitstellung von Geodaten steht die Spatial Database Engine SDE in direkter Verbindung zum Client.

Auch für die Nutzung von Geodaten und Analyseresultaten über das World Wide Web WWW ist SDD die Basis. Anfragen eines Browsers werden via WWW-Server und Internet Map Server IMS über das SDD beantwortet. Zur schnellen Bereitstellung von Geodaten steht die Spatial Database Engine SDE ebenfalls in Verbindung zum IMS.

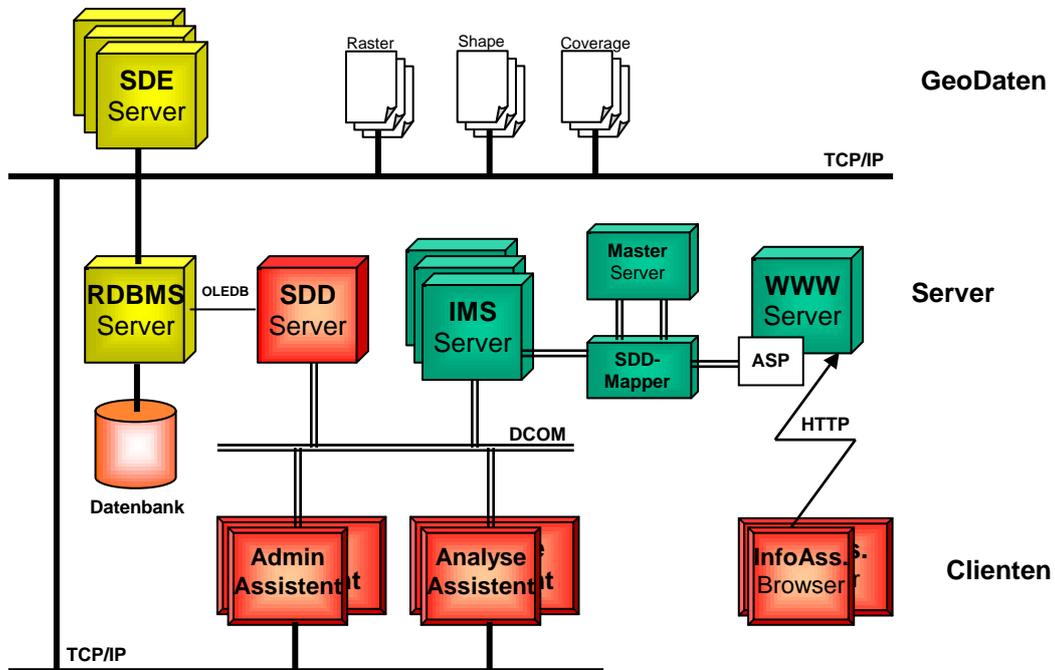


Abb. 9: Verteilte Client/Server-Architektur

### 5.3 Aufbau des SDD-Servers

Technisch betrachtet stellt der SDD-Server den Clients ein DCOM-Interface (Distributed Component Object Model) für die Applikationsobjekte zur Verfügung. Durch diese Technologie wird das SDD dem modernen Ansatz „Fat Server - Thin Client“ gerecht. Dabei holt sich der Client lediglich eine Referenz auf das sich im SDD-Server befindliche Applikationsobjekt, stösst eine Methode an, lässt den SDD-Server rechnen und kriegt das Resultat zurück. Die Clients können in einer beliebigen Programmiersprache entwickelt werden, wie Visual Basic, Delphi, Java, etc.

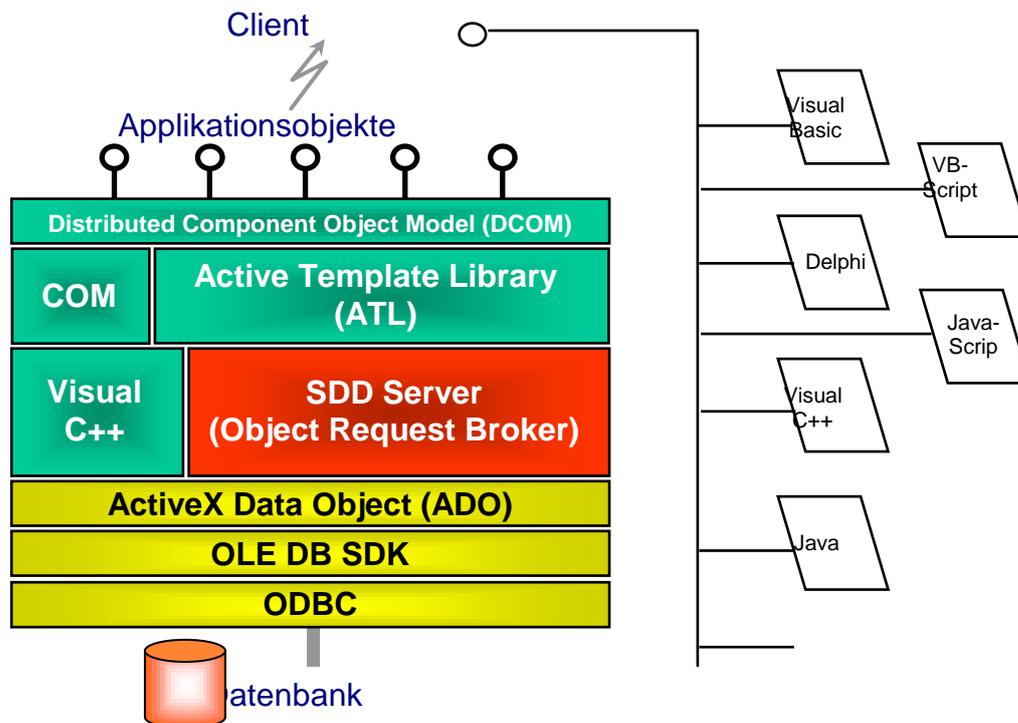


Abb. 10: Aufbau des SDD-Servers

Innerhalb von SDD sind als Systemkomponenten die permanente Datenhaltung und die Objekt-Instanziierung zu unterscheiden. Die permanente Datenhaltung basiert auf Datenstrukturen, die als Tabellenwerk auf einem RDBMS realisiert sind und die den Attributgehalt der Applikationsobjekte tragen. Die Instanziierung von Applikationsobjekten erfolgt im SDD-Server.

Die Applikationsobjekte werden zentral auf einem SDD-Server aus den RDBMS-Attributtabelle generiert und als instanziierte Objekte vorgehalten (Cache). Die Anbindung SDD zu RDBMS erfolgt über eine einzige (und damit leichter zu pflegende) ODBC-Schnittstelle auf dem SDD-Server. Die Applikationsobjekte werden über DCOM an die Applikationen transportiert. Hierfür wird lediglich TCP/IP als Verbindungsmedium vorausgesetzt sowie die Freigabe der entsprechenden Klassen-IDs auf dem SDD-Server. Die Applikationen empfangen vollständig instanziierte Applikationsobjekte mit erfolgter Methodenbindung.

## 5.4 Basis für GeoAssistenten

Aufbauend auf dem SDD wurden in Köln verschiedene GeoAssistenten konzipiert, die in Form einer Wertschöpfungskette den gesamten Informationsproduktionsprozess von der Datenmodellierung, über den Aufbau von Geodaten und deren Analyse bis zur Bereitstellung im WWW abdecken. Die GeoAssistenten werden mit Visual Basic unter Verwendung von MapObjects bzw. ArcObjects realisiert.

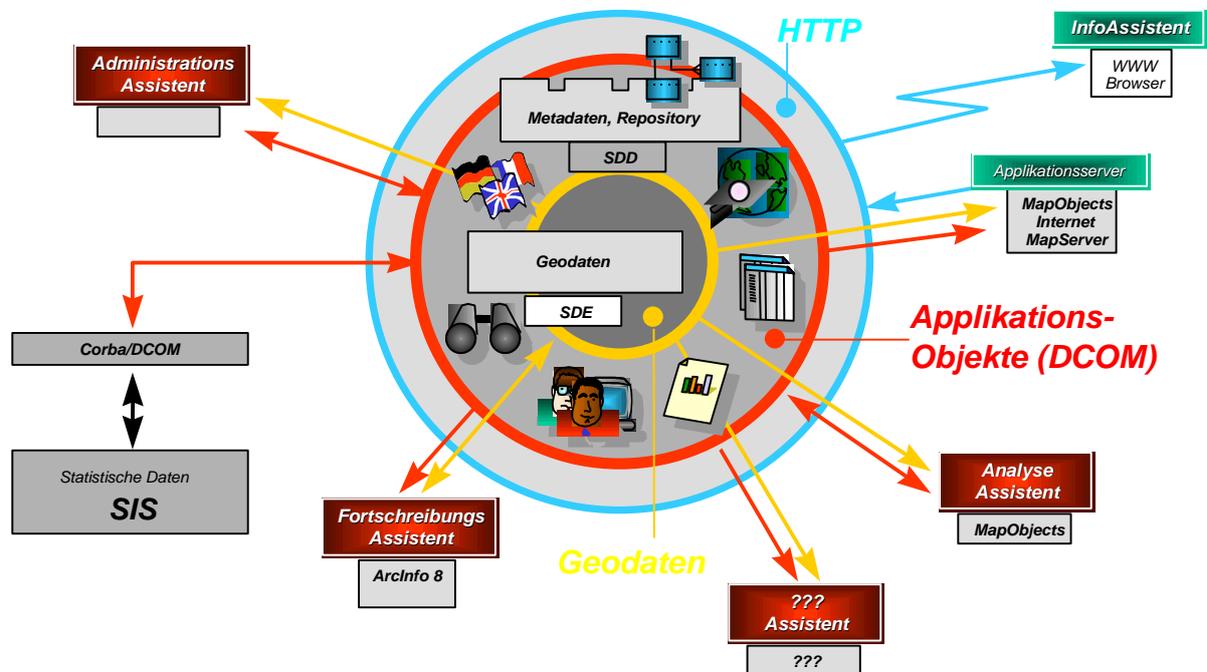


Abb. 11: SDD als Basis für GeoAssistenten

## 6 Komponenten für die Anwendungsprogrammierung

Das Semantic Data Dictionary stellt dem Anwendungsprogrammierer ein Interface zur Verfügung, mit dem schnell und einfach neue Anwendungen erstellt werden können. Dieses Interface basiert auf OLE/COM Objekten mit deren Methoden und Eigenschaften.

Komponenten sind für Entwickler gedacht, welche eigenständige Applikationen basierend auf dem SDD schreiben wollen. Komponenten sind COM-Controls (ActiveX/OCX,DLL) welche auf einer binären Ebene wiederverwendbar sind.

Bisher sind folgende Komponenten realisiert:

**MapManager:** Sie übernimmt die Steuerung der Karte und ist mit MapObjects 2 einsetzbar. Die Steuerung mit ArcInfo 8 ist in Planung für den FortschreibungsAssistenten.

**SetManager:** Sie ermöglicht das Speichern von Analyseergebnissen. Weiter ist diese Komponente die Schnittstelle zu einem Data Warehouse. Sachdaten aus dem Data Warehouse werden als Mengen gespeichert und können gemeinsam mit den Geobjekten ausgewertet werden. Der SetManager hat sogenannte OLAP-Funktionalität (Berechnen, Aggregation und Pivoting) integriert und kann mit Multidimensionalen-Attributtabelle umgehen.)

**QueryManager:** Sie dient für das Handling von Abfragen. Basierend auf den Beschreibungen des semantischen Datenmodells im SDD kann durch das semantische Datenmodell navigiert werden und entsprechende Verfahren/Abfragen formuliert werden.

**DataModel** Komponente: Sie bringt die Metadaten ins System.

Derzeit in Planung befindet sich eine **ThematicMappingManager** Komponente für das thematische Darstellen von Analyseergebnissen. Über diese Komponente können die Parameter einer thematischen Darstellung im SDD gespeichert werden. Thematische Darstellungen sind: Balken- und Kuchendiagramme, Klassengrenzen- oder Wertefärbungen, ...

Die Komponenten sind als Bausteine zu verstehen, die für bestehende und zukünftige Anwendungen wiederverwendet werden können.

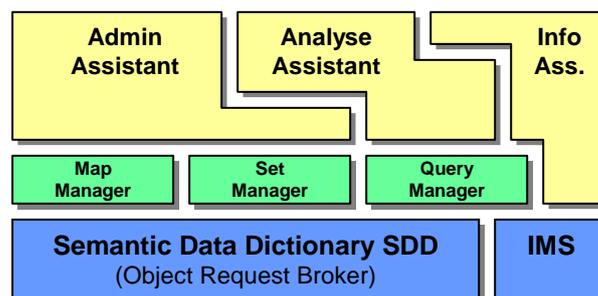


Abb. 12: Die Komponenten

## 6.1 MapManager

Der MapManager ist unterteilt in eine DLL- und eine OCX-Komponente. Die DLL-Komponente stellt ein vereinfachtes Interface zu SDD und zu einem „MapProvider“ für die Anwendungsprogrammierung zur Verfügung und übernimmt die Kartensteuerung z.B. zu MapObjects. Die OCX-Komponente ist das Interface (GUI) zum Anwender.

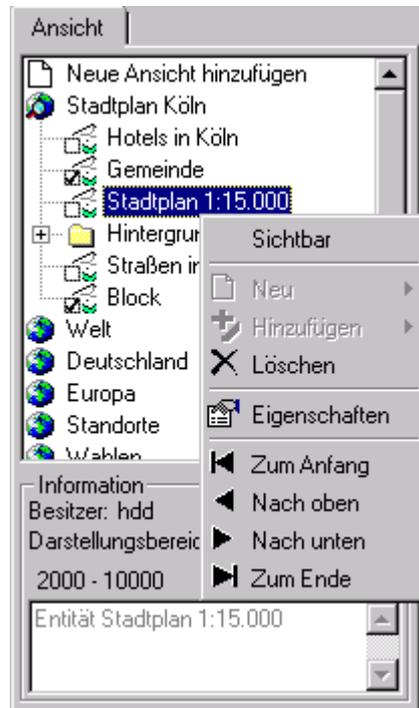


Abb. 13: MapManager-OCX

## 6.2 SetManager

Der SetManager ist unterteilt in eine DLL- und eine OCX-Komponente. Die DLL-Komponente stellt ein vereinfachtes Interface zu SDD für die Anwendungsprogrammierung zur Verfügung. Die OCX-Komponente ist das Interface (GUI) zum Anwender.

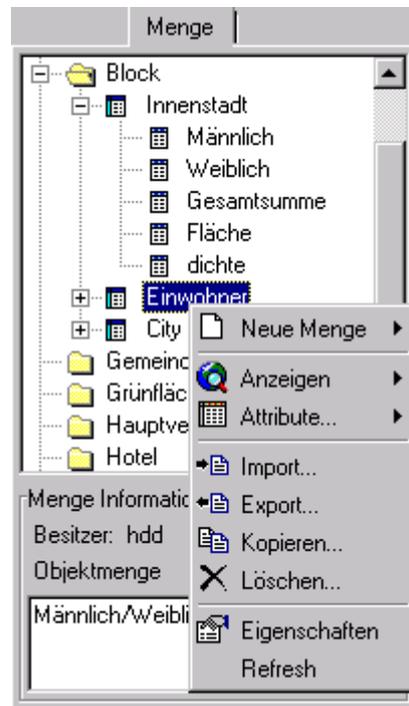


Abb. 14: SetManager-OCX

### 6.3 QueryManager

Der QueryManager ist unterteilt in eine DLL- und eine OCX-Komponente. Die DLL-Komponente stellt ein vereinfachtes Interface zu SDD für die Anwendungsprogrammierung zur Verfügung. Die OCX-Komponente ist das Interface (GUI) zum Anwender.



Abb. 15: QueryManager-OCX

## 7 Realisierungskonzept

### 7.1 Standards

#### 7.1.1 Objekt-Orientierung

Systemkomponenten werden als Objekte identifiziert, die durch ihre Klassen vollständig beschrieben und gesteuert werden. Alle Komponenten sind Objekte.

#### 7.1.2 Klassen, Klassenbegriff

Eine Klasse wird verstanden im Sinne einer formalen Vorschrift zur Instanziierung (Erzeugung) eines Typs eines Objekts. Ein Objekt ist ein gekapselter Verbund von Methoden und Eigenschaften (Ressourcen), der nach aussen hin keine unmittelbaren Seiteneffekte aufweist (Modulbegriff in strenger Form) und der in beliebiger Form mehrfach instanziiert werden kann. Es wird davon ausgegangen, dass die hier in allgemeiner Form beschriebenen Klasseigenschaften in allen relevanten Implementierungssystemen verfügbar sind. Dies bedeutet die *Definition* von Klassen, die *Kapselung* aller Klasseigenschaften und aller Klassenmethoden, und die gezielte Freigabe von Klassenmethoden durch Schnittstellen (*Interfaces*). Instanziierung und Mehrfachinstanziierung wird vorausgesetzt. *Vererbung* wird nur insofern beansprucht, als damit Typisierung innerhalb einer Klasse vermieden wird. Wenn Vererbung nicht möglich sein sollte, dann sind Redundanzen durch die Definition „ähnlicher“, wurzelloser Klassen eher zu dulden als Typisierung mit Fallunterscheidungen innerhalb einer Klasse. Bei der Portierung auf ein Klassensystem mit Vererbung sind „ähnliche“ Klassen leichter in eine Vererbungshierarchie überführbar als die Auflösung von Typisierungen innerhalb einer Basisklasse. Diese Forderung ist deshalb streng einzuhalten. Vererbung vermeidet die Redundanz paralleler, „ähnlicher“ Klassen und bedeutet den Gebrauch von *virtuellen Methoden*, wodurch *Polymorphie* ermöglicht wird. Mehrfachvererbung wird konzeptionell wegen der dadurch erzielbaren Eleganz verwendet, je nach Implementierungssprache (mit oder ohne Mehrfachvererbung) aber durch entsprechende Konstrukte auf der Instanzenebene realisiert bzw. simuliert.

Klassen sind in ihrer Implementierung zugleich Träger ihrer Primärdokumentation. Damit die Primärdokumentation von Klassen als solche bedeutsam sein kann, ist eine sprechende Namensgebung aller Bezeichner notwendig. Dies wird vorausgesetzt und ist damit Teil des Style-Guides.

#### 7.1.3 Kommunikation

Der Netz-Transport im LAN erfolgt über TCP/IP.

#### 7.1.4 Datenbank-Zugriff

Open Database Connectivity ODBC als Standard.

