

Objektorientierte Entwicklung von Planungswerkzeugen an Beispielen aus der Umweltplanung

Rainer HASELBERGER

(Dipl.-Ing. Rainer HASELBERGER, OstBR, Magistrat der Stadt Wien, Magistratsabteilung 14 - ADV, 1082 Wien, Rathausstraße 1;
e-mail: har@adv.magwien.gv.at)

1. SPEZIELLE PROBLEME DER EDV IN DER ÖFFENTLICHEN VERWALTUNG

Die Stadt Wien ist mit rund 60.000 Bediensteten eine der größten Verwaltungseinheiten in Österreich und aufgrund der Bundesverfassung, der Wiener Stadtverfassung und verschiedenster bundes- und landesgesetzlicher Materien in vielfältiger Art und Weise mit Planungs- und Umweltschutzaufgaben befaßt. Diese Aufgaben sind mithilfe der Elektronischen Datenverarbeitung bestmöglich zu unterstützen.

Aus der immensen Entwicklungsdynamik in der Informatik und bei den technischen Hilfsmitteln der Datenverarbeitung und der Kommunikation resultieren höchste Anforderungen an die Organisation, Planung und Durchführung des EDV-Einsatzes, die auch immer neue Überlegungen erforderlich machen, wie die Investitionen in die EDV gegen die rasche Entwertung durch immer kürzere Lebenszyklen gesichert werden können. Die öffentliche Verwaltung kann sich dem technischen Fortschritt nicht verschließen, da der Bürger die Information ebenso einfordert wie die möglichst rationale, effiziente und effektive Aufgabenwahrnehmung durch den Magistrat. Sie muß aber den Einsatz der EDV so organisieren, daß - ebenso wie bei der Nutzung der Umweltressourcen - auch bei der Nutzung der EDV-Ressourcen Nachhaltigkeit erzielt werden kann.

Ein ganz wesentliches Problem für die öffentliche EDV-Verwaltung bildet ebenso wie für jeden Privaten der hohe Weiterbildungsbedarf der Mitarbeiter, da das bestehende Wissen in vielen Bereichen (zum Beispiel bei den Office-Paketen) eine Halbwertszeit von unter einem Jahr aufweist. Ein weiteres Problem ist der wachsende Koordinierungsbedarf, der sich aus dem steigenden Daten- und Informationsangebot ergibt, um zu verhindern, daß ein und dasselbe Problem mehrfach unterschiedlich, womöglich noch inkompatibel gelöst wird. Ein drittes Problem ergibt sich aus der Entwertung von Entwicklungsinvestitionen, da viele Werkzeuge, die heute selbst entwickelt werden müssen, morgen bereits in Standardapplikationen zu einem Bruchteil des Preises der Eigenentwicklung verfügbar sind.

Gerade die Verwaltungsfelder Planung und Umwelt sind von dieser Informationsexplosion in besonderer Weise betroffen, da erst durch den Einsatz der oben angeführten Hilfsmittel viele der Aufgaben dieser Gebiete bearbeitbar und erfüllbar werden. Sie zeichnen sich neben diesen allgemeinen Problemen noch durch spezielle Eigenheiten aus, die den EDV-Einsatz weiter komplizieren.

1.1. Organisatorische Probleme des EDV-Einsatzes in Planung und Umwelt

Wie jedes öffentliche Verwaltungshandeln erfolgt auch die Verwaltung und Planung der Umwelt auf der Grundlage der vom Gesetzgeber erlassenen Gesetze. Planung und Umweltschutz sind komplexe Materien. Das heißt, sie sind in den unterschiedlichsten Gesetzen geregelt und werden von verschiedenen Institutionen auf unterschiedlichen Verwaltungsebenen (Bund, Länder, Gemeinden) und in unterschiedlicher Art und Weise vollzogen, woraus sich Überschneidungen bzw. Kommunikationsprobleme ergeben.

Ein weiteres Problem ist darin zu sehen, daß der Gesetzgeber in vielen Gesetzen auf die Möglichkeiten und den Einsatz der EDV in der Verwaltung keine Rücksicht genommen hat (was in den meisten Fällen mit dem Alter der Gesetze zu erklären ist). Damit fehlen einerseits in manchen Bereichen die legislativen Grundlagen für einen EDV-Einsatz völlig. Andererseits sind dort, wo technische Vorschriften existieren, diese unter Umständen veraltet und behindern damit eine Datenverarbeitung nach modernen Gesichtspunkten eher als sie zu fördern. Dazu kommt gerade bei der Finanzierung der Investitionen die schwierige Zuordnung und Quantifizierung des Nutzens, der sich aus den EDV-Investitionen in diesen Bereichen ergibt, die aber aufgrund der Finanzierungsprobleme der öffentlichen Budgets immer größere Bedeutung erhält. Gerade im Verteilungskampf um die knappen Mittel geraten dadurch Planung und Umwelt gegenüber anderen Verwaltungsbereichen, die kurzfristigere Renditen versprechen, ins Hintertreffen.

Entsprechend der Geschäftseinteilung und Geschäftsordnung des Magistrats ist derselbe in Zuständigkeitsbereiche gegliedert, die von verschiedenen Organisationseinheiten (Magistratsabteilungen = MA) verwaltet werden, zwischen denen nicht zuletzt aufgrund der räumlichen Zersplitterung und unterschiedlicher technischer Ausstattung der Informationsaustausch nicht immer im erwünschten Ausmaß gegeben ist. Auch die heutigen Methoden der Telekommunikation haben noch nichts daran geändert, daß viele der Daten, die in einer Abteilung erhoben und aktualisiert werden, anderen Abteilungen nicht zur Verfügung gestellt werden - selbst wenn das Datenschutzgesetz und die Materiegesetze dies zuließen.

1.2. Technische Probleme

Aus der technischen Natur der Planungs- und Umweltdaten gibt es eine erkleckliche Menge an technischen Problemen, von denen nur die wichtigsten hier cursorisch angerissen werden sollen:

- Planungs- und Umweltdaten sind äußerst **vielfältig**. Es gibt Daten über die Luft, den Boden, das Wasser, das Gestein, die Tier- und Pflanzenwelt und nicht zuletzt über den Menschen und alle seine Eingriffe in die Umwelt. Um hier die richtigen und relevanten Informationen zu finden und sie nutzen zu können, bedarf es entsprechender Werkzeuge.
- Daten ohne **Kontextinformationen** sind wertlos - ein Meßwert, dessen Meßparameter, Ort, Meßverfahren, Zeitpunkt, Genauigkeit, Umweltbedingungen und Methode unbekannt sind, ist nicht interpretierbar. Der operative Bearbeiter weiß in der Regel über alle diese Kontextvariablen bescheid (vor allem, wenn die Messung manuell durchgeführt wird) und scheut den Zusatzaufwand der Erfassung dieser Informationen zum eigentlichen Inhalt. Ohne sie ist das Datum aber für keinen anderen interpretierbar.
- Die Fachleute der Planungs-, Umwelt- und Geowissenschaften sprechen verschiedene Fachsprachen, die zu einer semantischen Verwirrung führen, für die es heute noch keine Lösung gibt.
- Die Umwelt ist ein komplexes Wirkungsgefüge, dessen Zusammenhänge stark vernetzt und erst zu einem sehr kleinen Teil erkannt und beschrieben, aber kaum noch auf Zusammenhänge hin analysiert wurden. Die EDV braucht - noch - klare Regeln und Algorithmen, wie sie die Umweltwissenschaften nur vereinzelt bieten können.
- Die Daten fallen in großen Mengen an - bei einer Meßstelle, an der 5 Parameter in 5-minütigen Intervallen gespeichert werden, entstehen pro Tag 1440 Datensätze, und im Jahr 525.600.
- Durch die organisatorische Vielfalt, die bestehenden Insellösungen, die semantischen Probleme und die großen Datenmengen ergibt sich auch ein technisches Kommunikationsproblem.
- Und nicht zuletzt unterscheiden sich Planungs- und Umweltdaten von anderen Daten durch den Raumbezug der Objekte und Attribute, der unbedingt abgebildet werden muß, dessen Darstellungswerkzeuge (GIS) erst am Anfang stehen, und deren Gebrauch noch nicht Allgemeingut ist.
- Die gesetzlich geregelte Kommunikation mit dem und Partizipation des Bürgers (Umweltinformationsgesetz, Umweltverträglichkeitsprüfungen,...) erfordert eine besondere, arbeitsintensive redaktionelle (Auswahl, Such und Interpretationshilfen), administrative (Berechtigungen) und multimediale Aufbereitung der Daten, die über die dem Fachpublikum zumutbare Präsentation weit hinausgeht.

1.3. Anforderungen aus einigen laufenden Applikationsprojekten

Anbei zur Illustration eine kurze Beschreibung einiger derzeit laufender Projekte, die exemplarisch für die genannten Anforderungen sind:

1.3.1. WUIS - Wiener Umweltinformationssystem: Ein umfassendes Informationssystem

Das WUIS ist ein Datawarehouse unter ORACLE mit GIS-Einbindung (Arc/Info), in dem in einer Sekundärdatenbank die Umweltdaten aus den operativen Datenverarbeitungssystemen der Umweltabteilungen gesammelt werden. Über eine durch die Stadt Wien entwickelte Client-Applikation (ESRI-ArcView 2, MS Visual Basic 3, MS Access 2, ODBC) kann der Benutzer auf die Daten zugreifen, über Metadaten Informationen suchen, mit MS Access auswerten und in andere MS Office-Produkte einbinden. Die erste Release des WUIS, die zur Zeit in Produktion ist, hat gezeigt, daß es aus Gründen der

Wiederverwendbarkeit des Codes notwendig sein wird, bei der Umstellung des Clients auf 32bit-Applikationen (VB 4, MapObjects, MS Access 7,...) besser gekapselte Komponenten zu erzeugen, die auch außerhalb des WUIS eingesetzt werden können. Im Wesentlichen bauen alle den folgenden Aussagen zugrundeliegenden Erfahrungen auf dem Projekt WUIS auf. Es ist geplant, die Umstellung des WUIS 1997 durchzuführen.

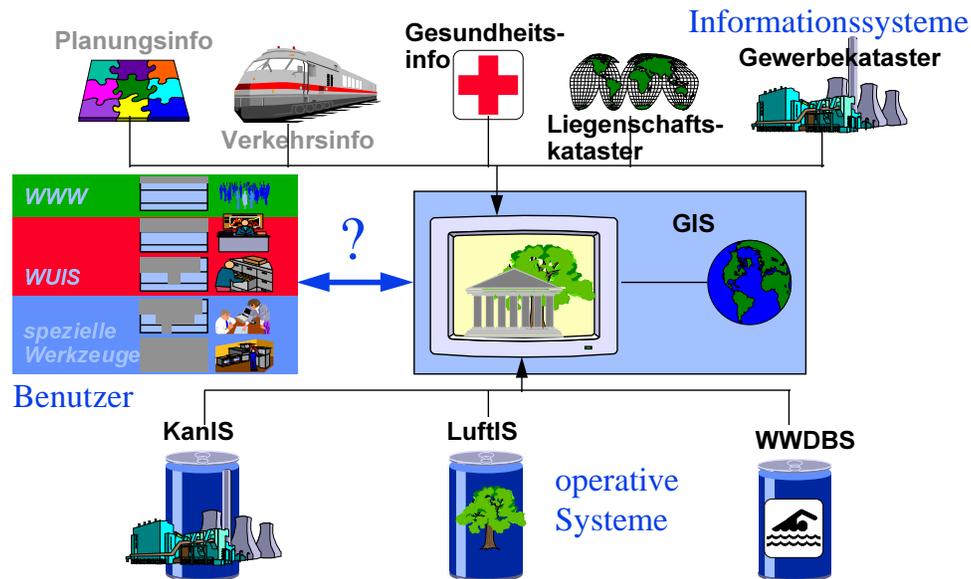


Abb. 1: WUIS - Wiener Umweltinformationssystem

1.3.2. Der Baulückenkataster: Ein Pilotprojekt für neue Technologien

Der Baulückenkataster der Stadt Wien ist zur Zeit eine MS Access 2-Client/Server-Applikation, in der etwa 1500 Baulücken im dichtverbauten Stadtgebiet bei der MA 40 - Technische Grundstücksangelegenheiten verwaltet werden. Diese Daten stehen dem Wiener Bodenbereitstellungs- und Stadterneuerungsfonds im Rahmen eines Auftrags der Stadt Wien ebenso zur Verfügung wie den Planungsdienststellen und privaten Interessenten. Er enthält zur Zeit nur tabellarische Ausgabemöglichkeiten und Photos, aber keine GIS-Anbindung. Es ist geplant, im Rahmen eines Pilotprojekts Anfang 1997 den Baulückenkataster auf die 32bit-Plattform umzustellen, ein GIS einzubinden und dabei die im Folgenden beschriebenen Techniken auf ihre Brauchbarkeit zu testen.

1.3.3. WWDBS - Wasserwirtschaftliches Datenbanksystem: Integration von tabellarischen Daten und GIS

Das Wasserwirtschaftliche Datenbanksystem der MA 45 - Wasserbau ist zur Zeit im Detailkonzeptstadium. Es soll alle Daten aus dem Wasserbaubereich der Stadt Wien, angefangen bei den Wasserrechten über die Grundwasser- und Oberflächenwasserdaten bis zu den Altlasten gemeinsam verwalten und dem Benutzer Zugriffswerkzeuge auf diese Daten bieten, mit deren Hilfe er auch räumliche Auswertungen und Darstellungen erzeugen kann. Auch bei der Datenerfassung und -änderung ist die GIS-Unterstützung erforderlich. Dieses Projekt ist relativ komplex und deshalb in mehrere Teilprojekte zerlegt, deren erstes neben einer zentralen Objektverwaltung die Wasserrechte umfassen und 1997/98 verwirklicht werden soll.

1.3.4. Kanalmanagement und -controlling: Betriebsinformationssystem

Zum zeitgemäßen Betriebsmanagement der Abwasseranlagen der MA 30 - Kanalisation und zur Investitionsplanung ist es erforderlich, die bestehenden Anlagen und deren Betriebszustände sowie Ausbaupläne zu erfassen, zu bewerten und Betriebsfälle zu simulieren. Daneben ist es erforderlich, die im Betrieb anfallenden Personal- und Materialaufwände zu dokumentieren und zu bearbeiten, sowie die für die Verrechnung der Gebühren notwendigen Datengrundlagen zu schaffen und zu verwalten. Dieses Projekt sollte bereits 1996 starten, wurde jedoch aufgrund absehbarer organisatorischer Veränderungen auf 1997 verschoben. Auch hier ist es notwendig, räumliche Informationen eng mit Sachdaten zu verknüpfen und den Benutzern ein möglichst mächtiges Auswertungs- und Darstellungswerkzeug zur Verfügung zu stellen.

1.3.5. Verkehrsorganisationsprojekte - Individualverkehr: Integration und Workflowmanagement

Im Bereich der MA 46 - Verkehrsorganisation gibt es eine große Zahl von Aufgaben, die die Verwaltung raumbezogener Ressourcen erfordern: Verkehrszeichen, Bodenmarkierungen, Radwege, Unfälle... Teilweise werden noch Handkarteien geführt, andererseits werden die Ressourcen in lokalen Inselanwendungen durch die jeweilige Unterorganisationseinheit so verwaltet, daß die anderen Teileinheiten keinen Zugang zu den Daten haben können. Es wird eine der wesentlichen Aufgaben der nächsten Jahre sein, für die MA 46 eine gemeinsame Datenbasis aufzubauen und für die Sachbearbeiter die geeigneten, leicht benutzbaren Werkzeuge zur Erfassung, Verarbeitung und Darstellung der Daten zu schaffen. In diesen Projekten stellt neben den oben angesprochenen Problemen die Abbildung der Geschäftsabläufe im Rahmen eines Workflowmanagements eine besondere Herausforderung dar.

Alle genannten Anwendungen entstehen parallel und organisatorisch weitestgehend unabhängig voneinander. Die Aufgabe der EDV-Abteilung der Stadt Wien (MA 14 - ADV) ist es, dafür zu sorgen, daß die darin enthaltenen Synergien, vor allem was Daten und Software betrifft, genutzt werden.

2. DIE LÖSUNG: COMPONENTWARE

Einen Ausweg aus dem Dilemma zwischen begrenzten Ressourcen und zunehmender Quantität und Komplexität der Datenverarbeitung scheint der bereits seit den Achtzigerjahren in Ansätzen entwickelte Weg darzustellen, die EDV in Komponenten zu zerlegen, die über genormte Schnittstellen zusammengefügt werden können. War dieser Ansatz der objektorientierten Datenverarbeitung bisher eher ein akademisches Forschungsfeld, so werden nun nach und nach kommerziell einsetzbare Produkte verfügbar, die unter anderem bei der Stadt Wien zum Einsatz kommen sollen. Der vorliegende Artikel kann angesichts des rasanten Fortschritts nur eine Momentaufnahme des Einsatzstandes für einige in Entwicklung befindliche Projekte darstellen und erhebt weder Anspruch auf Vollständigkeit noch auf Dauerhaftigkeit.

Dieser Ansatz ist aber weitgehend unberührt von der fachlichen Diskussion, ob das künftige Standardendgerät WS, NC oder PC genannt werden wird, ob die Sprache Java, HTML, Visual Basic, SMALLTALK oder C++ heißt, und ob das zugrundeliegende Modell OLE, ActiveX, OpenDOC oder CORBA gerufen werden wird. Allen diesen Kürzeln ist gemeinsam, daß sie auf dem neuen Konzept aufsetzen, das objektorientierte Programmierung heißt. Dem Paradigma der Objektorientierung in der Programmentwicklung steht gleichwertig die objektorientierte Datenbank gegenüber, die aber abweichend von den Entwicklungswerkzeugen noch nicht jene Marktreife erlangt hat, die einen breiten kommerziellen Einsatz rechtfertigen würde, da weder die Standards noch die Produkte in ausreichender Reife verfügbar sind.

Mit der Beschreibung der Objektorientierten Programmierung setzen sich viele Schriften auseinander, die hier nicht ausführlich zitiert werden sollen. Einen wesentlichen Beitrag hat die Object Management Group geleistet, in deren Papers man die entsprechenden Grundlagen nachlesen kann [Object Management Group 1995].

2.1. Was ist ein Objekt?

Ein Objekt ist ein abgrenzbarer Gegenstand der wirklichen Welt oder der Erkenntnis, der eindeutig identifiziert werden kann. (In Wirklichkeit nur ein rechnergebundenes, abstraktes Abbild eines Gegenstandes!!) Es verfügt über Eigenschaften, die es von anderen Objekten unterscheidet und kann mit bestimmten Methoden manipuliert werden¹. Gleichartige Objekte können in Kategorien (Objektarten, Typen, Objektklassen) eingeteilt werden, die prototypisch die Eigenschaften der Objekte enthalten, und deren Instanzen die einzelnen Objekte sind. Weder physische Ausprägung noch irgendwelche andere spezielle Voraussetzungen (zB Edv-technische Verwendbarkeit) sind erforderlich, um einen Gegenstand zum Objekt zu machen. Vielmehr ist - wie bei der Auswahl der Variablen in einem mathematischen Modell - alleine die problemadäquate Opportunität dafür ausschlaggebend, ob in einem bestimmten Applikationsmodell ein Gegenstand als Objektklasse deklariert wird, oder ob er nur eine spezielle Ausprägung einer anderen Objektklasse darstellt. So gibt es sicherlich Gründe, in manchen Anwendungen

Hauptstraßen und Anliegerstraßen als eigene Objektklassen zu definieren, oder Überland- und Stadtstraßen, Bundes-, Landes-, Gemeindestraßen,... - für viele Aufgabenbereiche wird jedoch einfach die Objektklasse Straße - versehen mit dem Attribut „Straßenkategorie“ - ausreichen.

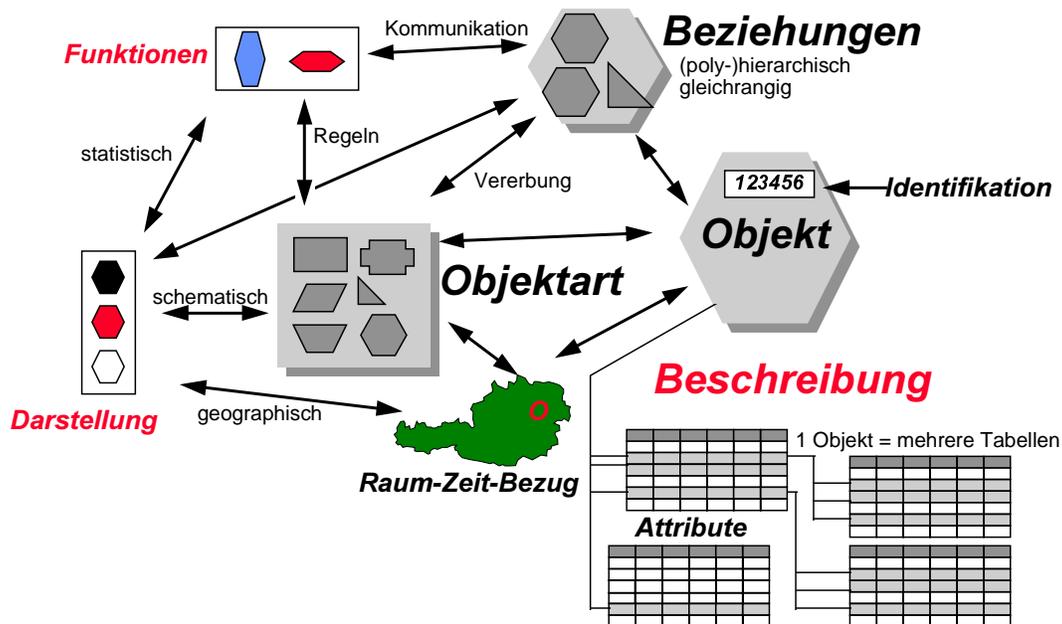


Abb 2: Objekt

Wie die Erfahrung zeigt, empfiehlt es sich, bei der Bezeichnung der Objektklassen möglichst umgangssprachliche, allgemeinverständliche Begriffe zu verwenden und - auch unter bewußtem Verzicht auf technisch sinnvolle Abweichungen - die Definition der Objektklasse möglichst dem sprachlichen Usus anzupassen. Denn nur in diesem Fall kann der größte Vorteil der objektorientierten Denkweise für Entwurf, Programmierung und Datenhaltung genutzt werden - die Nähe zu unserer Sprach- und Gedankenwelt.

In vielen Fällen lassen sich synthetische Objekttypen nicht vermeiden, die den Benutzer mehr irritieren als zum Verständnis der abgebildeten Wirklichkeit beitragen. So war es im WUIS notwendig, da Arc/Info nur Coverages mit reinen Geometriedimensionsklassen (Point, Arc, Polygon) zuläßt und außerdem disjunkte Objekte innerhalb einer Coverage zwingend erfordert, daß aus den Gewässern die Objekttypen „Flächige Gewässer“ und „Linienförmige Gewässer“ erzeugt werden mußten, oder daß es eine Objektklasse „Naturdenkmal“ gibt, die aber räumlich nur in ihren Derivaten „Naturdenkmalpunkt“ und „Naturdenkmalfläche“ dargestellt werden können. Des Weiteren mußten kontinuierliche Phänomene (zB Temperatur, Vegetationszeit) als eigene Objektklasse Image definiert werden, da Isochronen und Flächen gleicher mittlerer Werte keine sinnvollen Objekte im Sinne der Objektterminologie darstellen. Interessanterweise ist die im OpenGIS-Guide [OpenGIS Consortium 1996] enthaltene Lösung ähnlich der, die unabhängig davon für das WUIS erdacht werden mußte. Die unten angeführten neuen Entwicklungen im GIS-Bereich werden aber dazu führen, daß viele dieser sperrigen Begriffe in der nächsten WUIS-Release wieder verschwinden werden.

Ein Objekt gibt seine Eigenschaften nur über die offengelegten, hoffentlich ausreichend genau beschriebenen und stabilen Schnittstellen, die Kapsel, preis. Die Eigenschaften der Objekte enthalten neben den Datenattributen auch die Methoden, mit deren Hilfe das Objekt manipuliert (verändert, erzeugt, gelöscht) werden kann. Ob und in welcher Form die Daten vor dieser Preisgabe gespeichert sind, oder ob sie erst zum Zeitpunkt der Abfrage generiert werden, ist für den Benutzer des Objekts nicht erkennbar. Das heißt, daß der Benutzer bei einer echten objektorientierten Anwendung nicht weiß, ob seine Anforderung an das Objekt eine „echte“ (Daten-)Eigenschaft, oder eine Methode abfragt.

¹ OMG 1995: „...is an identifiable, encapsulated entity that provides one or more services that can be requested by a client...“

2.2. Programmentwicklung

Für die Objektorientierte Programmentwicklung stehen heute neben hochentwickelten Programmiersprachen (C++, Smalltalk,...) und einfacheren Ansätzen (Visual Basic 4, Java) auch Entwicklungswerkzeuge zur Verfügung (sprachensensitive Editoren, Debugger, Performancetestwerkzeuge, Codeverwaltungssysteme,...), die eine rasche und problemlose Programmentwicklung ermöglichen, wobei die Qualifikationsanforderungen an die Entwickler/Prototyper abgestuft werden können und eine brauchbare Produktivität erreicht wird.

Object Request Broker (worunter so unterschiedliche Produkte wie CORBA, ORBIX oder OLE, ActiveX fallen) sind genormte (zB in der Sprache IDL - Interface Definition Language der Object Management Group [Object Management Group 1996] geschriebene) Verzeichnisse von Objektklassen und deren Schnittstellen(-beschreibungen), über die sich ein Benutzer (bzw. das von ihm benützte Programm) ein Verzeichnis der verfügbaren Objekte und Eigenschaften und Methoden erstellt oder automatisch ein Objekt/eine Eigenschaft aktivieren lassen kann, unabhängig davon, auf welchem Rechner eines Netzwerks dieses „Service“ angeboten wird.

Neben diesen grundlegenden Konzepten der Objektorientierung gibt es noch weitere Konzepte, die die Arbeit mit Objekten vereinfachen:

- Die Vererbung von Eigenschaften zwischen Objektklassen (Generalisierung, Spezialisierung), verbessert die Wiederverwendbarkeit des Codes.
- Zwischen Objekten unterschiedlicher Klassen oder der gleichen Klasse können hierarchische oder assoziative Verbindungen bestehen, die über entsprechende Datenverbindungen und Funktionen abgebildet werden müssen.
- Polymorphismus, Overloading: verschiedene Objektklassen können gleichnamige Eigenschaften haben, die unterschiedlich implementiert sind (zB Rectangle.Draw(pt1, pt2) z.U.v. Circle.Draw(pt, r)).
- Im Bereich der raumbezogenen Datenverarbeitung kommen zusätzlich noch Eigenschaften aus der Geometrie und der Topologie der Objekte hinzu, die allgemein abgebildet werden könn(t)en.

Diese Konzepte werden in unterschiedlichem Ausmaß von Entwicklungssystemen unterstützt.

Was daneben aber bei allen Entwicklungswerkzeugen fehlt, ist die direkte Einbindung von Dokumentations- und Hilfgenerierungswerkzeugen. Die Erstellung eines ausreichend komfortablen Hilfesystems für eine Applikation ist essentiell für die ökonomische Nutzbarkeit, erfordert aber unserer Erfahrung nach mit den bestehenden Werkzeugen 60-70% der Entwicklungs- (Detailkonzept + Prototyp + Codierung) zeit. Dieser Aufwand wird sich, sollten sich die Werkzeuge nicht verbessern, zukünftig durch die steigenden Erwartungen der Benutzer noch erhöhen.

2.3. Datenhaltung

Das Konzept der Kapselung (encapsulation), das in der Programmentwicklung (u.a. auch in Java, C++, Visual Basic 4) bereits weitgehend verwirklicht ist, steht im Bereich der Datenspeicherung noch nicht im gleichen Umfang zur Verfügung. Das liegt v.a. darin begründet, daß die persistente Vorhaltung von identifizierbaren Objekten auf einem Datenträger und die spätere Neuidentifikation in einem anderen Kontext wesentlich komplexer ist als die Generierung, Verwendung und schließliche Zerstörung eines Objekts in einem kontinuierlichen Prozeß.

Objektorientierte Datenbanken gibt es für den operativen Einsatz noch nicht mit entsprechendem Kosten-/Leistungs-/Sicherheitsverhältnis wie relationale Datenbanken. Auch fehlt für den objektorientierten Datenzugriff eine genormte Datenabfragesprache vergleichbar mit SQL. Meiner Meinung nach gehört zu einer objektorientierten Datenbank neben der Verwaltung der Objektdaten auch die Verwaltung des Codes der Objektmethoden und der zugehörigen Metadaten (Beschreibungen, Dokumentationen, Navigatoren...). Auch diese Funktionalität gibt es zur Zeit noch nirgendwo in einer der reinen Softwareerstellungswerkzeugen vergleichbaren Qualität. Aus den genannten Gründen wird es auf absehbare Zeit notwendig sein, will man nicht auf die Objektorientierung verzichten, durch programmatische Überbauten über die relationalen Datenbankstrukturen so etwas wie eine objektorientierte Datenbank zu erzeugen.

Für die Vorhaltung und den Zugriff auf Metadaten, die aufgrund der explodierenden Informationsmengen gerade in den Bereichen Umwelt und Planung besondere Bedeutung zukommt, werden zur Zeit verschiedenste Konzepte entwickelt, wobei auch Normungsbestrebungen bestehen, die aber nur für Nischen Ergebnisse erbracht und noch zu keinen verbindlichen und allgemein anwendbaren Werkzeugen geführt haben.

2.4. Geographische Daten

Bisher wurden geographische Daten vor allem layerbezogen, topologisch aufbereitet vorgehalten und mußten mühsam mit den Relationen oder objektorientierten Paradigmen der Datenbank- und Entwicklungssysteme in Einklang gebracht werden. GIS-Entwicklungssysteme waren meist völlig von anderen Entwicklungssystemen getrennt und nur über proprietäre Sprachen (wie AML, Avenue,...) zugänglich. Die Datenhaltung erfolgte in proprietären Fileformaten (Coverage, DWG, ...), die nur über die proprietären Sprachen zugänglich waren und Legionen von Konvertierungsvorgängen in alle Richtungen erforderlich machten.

Neue Ansätze im GIS-Bereich lassen hoffen, daß sich diese Sprachverwirrung in Zukunft vermindern wird:

- Die ÖNORM A 2260 (und A 2261) stellt einen beachtlichen - wenn auch möglicherweise durch europäische und internationale Ansätze überhöhten - Versuch dar, eine genormte GIS-Schnittstelle unter Einsatz des objektorientierten Paradigmas festzulegen.
- Das OpenGIS-Konsortium beschäftigt sich ebenfalls mit der Normierung von GIS-Services und -daten.
- Die Firma ESRI bietet seit kurzem mit dem Produkt SDE einen funktionellen (Index-)Überbau über relationale Datenbanken an, der eine objektorientiertere GIS-Datenverarbeitung erlaubt als die bisherigen Speicherungsverfahren. Gleichzeitig wird mit MapObjects ein Objektwerkzeugkasten angeboten, der den Zugriff auf diese Daten aus Standardapplikationen mit den dort verfügbaren Entwicklungswerkzeugen erlaubt. Diese Verfahren befinden sich zur Zeit in der Evaluation. Es ist zu hoffen, daß sie sich in Zukunft bei der Erstellung wiederverwendbarer Programmkomponenten bewähren werden.

Daneben können im wissenschaftlichen Bereich viele Ansätze zur Implementierung von objektorientierten spatio-temporalen GIS verfolgt werden, deren Übernahme in den kommerziellen Einsatz aber noch einige Jahre dauern wird. [Luttermann H. 1996] gibt eine gute Zusammenstellung von spatio-temporalen Datenarten, die ein derartiges GIS verarbeiten können muß (nach Luttermann, Beispiele):

| | | | |
|-----------------------|--------------------|-----------------------|------------------------|
| Zeit\Raum | Einzelobjekt | geordnete Objektmenge | Kontinuum |
| Einzelobjekt | Punkt/Linie/Fläche | topologische Coverage | Oberfläche, Image |
| geordnete Objektmenge | Objekt-Historie | Coverage-Historie | Frame-/Slide-Set |
| Kontinuum | Ganglinie am Pegel | Stoffflüsse in Netzen | Animation, Ausbreitung |

3. VORGEHENSMODELL FÜR OO ANWENDUNGEN

Das objektorientierte Paradigma macht neben dem Einsatz neuer Werkzeuge für Datenhaltung und Programmentwicklung auch die Einführung einer neuen Projektkultur erforderlich. Zusätzlich zu den bisherigen Schritten im Vorgehensmodell und den dabei erzeugten Dokumenten sind weitere Schritte und Konzepte erforderlich, um die Wiederverwendbarkeit der erzeugten Komponenten und Verfahren zu gewährleisten. Zumeist gelingt es erst im Rahmen mehrerer Iterationen - im besten Fall noch im Rahmen eines begleiteten Prototyping - suffiziente Objektklassen mit allgemein verwendbaren Eigenschaften und Methoden zu definieren:

Rollen

Identifikation der Personen

Identifikation, Beschreibung der Abläufe und Verfahren: Administration, Datenerfassung, Auskünfte

Geschäftsfälle und Rechte: Standardabläufe, Rollen

Festlegung der Ausstattung der Arbeitsplätze mit Hardware

Daten

Datenmodell: Tabellen, Attribute, Relationen

Einrichtung der Testdaten(bank)

Applikationen, **Services, OLE-Server**

Objekte

Eigenschaften

Funktionen, Methoden

Prototyping und Tests in einer Testshell

Entwicklung

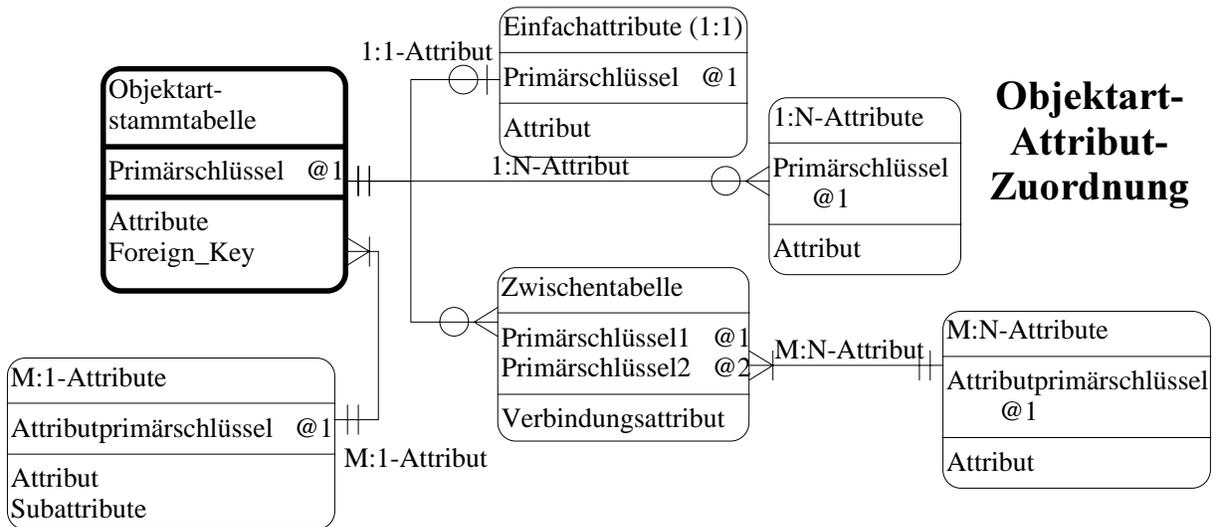
Die beiden folgenden Absätze sollen ganz kurz skizzieren, wie Beispiele für die genannten Dokumente aussehen könnten:

3.1. Datenmodellierung für GIS und RDBMS

In Erweiterung zum gebräuchlichen Entity-Relationship-Modell müssen für den objektbezogenen Überbau über die relationale Datenbank weitere Informationen im Datenmodell festgehalten werden: Vererbungen, Objektartstammtabellen usw. Diese Erweiterung des Entity-Relationship-Modells ist aber relativ einfach und in der folgenden Abbildung dargestellt. Zur Beschreibung des Datenmodells wird folgende Syntax gewählt (Objektart-Stammtabellen (=Objektarten) sind fett umrandet):

Größeren Aufwand erfordert die Erfassung dieser Daten in Metatabellen der Metadatenbank, wofür es geeigneter Werkzeuge bedarf, die heute noch eigens zu entwickeln sind.

Des Weiteren erforderlich ist die Erstellung von Routinen, um diese Metastruktur nutzen zu können. Der Benutzer will in der Regel hierarchisch oder über Objektverbindungen auf die Objekteigenschaften zugreifen können, ohne die relationale Struktur im Detail zu kennen. Er will Objekte nach ihren Eigenschaften suchen und manipulieren können, egal, ob die jeweilige Eigenschaft in der Stammtabelle der Objektart enthalten ist, oder in einer m:n-Relation zum Objekt steht (s. Abb. 3).



Objektart-Verbindungen

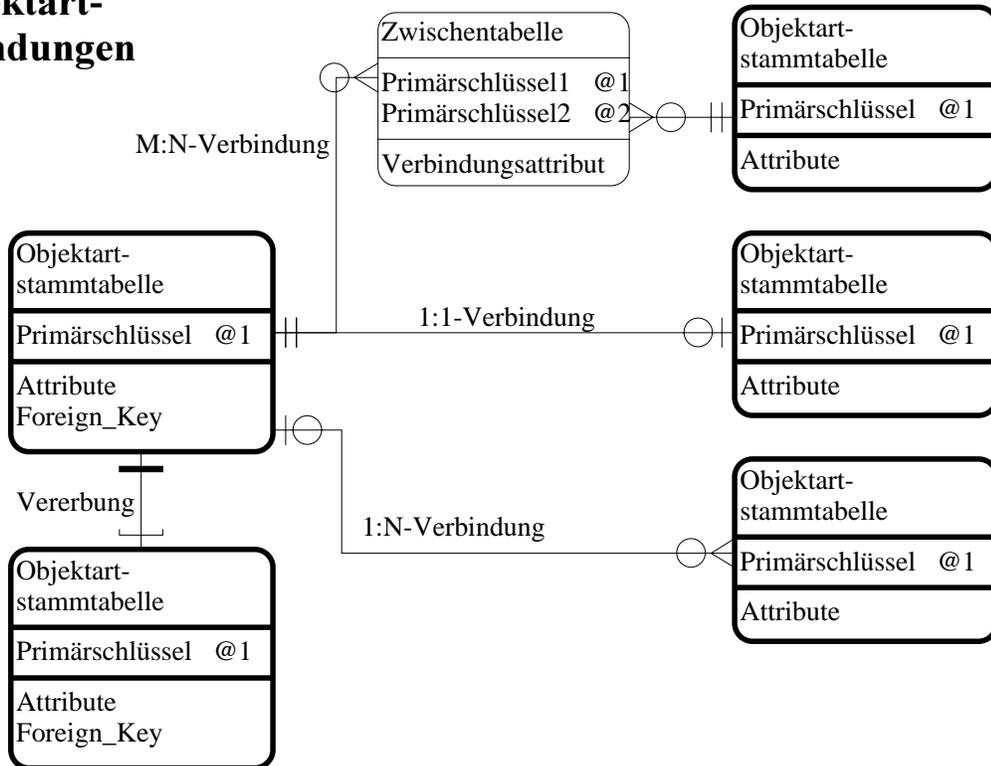


Abb 3: objektbezogenes Datenmodell - Legende

Die Abb. 4 auf der nächsten Seite zeigt als Beispiel das Datenmodell für die Objektarten Straßenabschnitt, Straße, Kreuzung, Adresse, Baublock, Grundstück, Nutzung,... In diesem Teilmodell gibt es klarerweise Verweise auf Objektarten (zB Gemeindebezirk), die in diesem Teil des Datenmodells nicht vollständig angeführt sind.

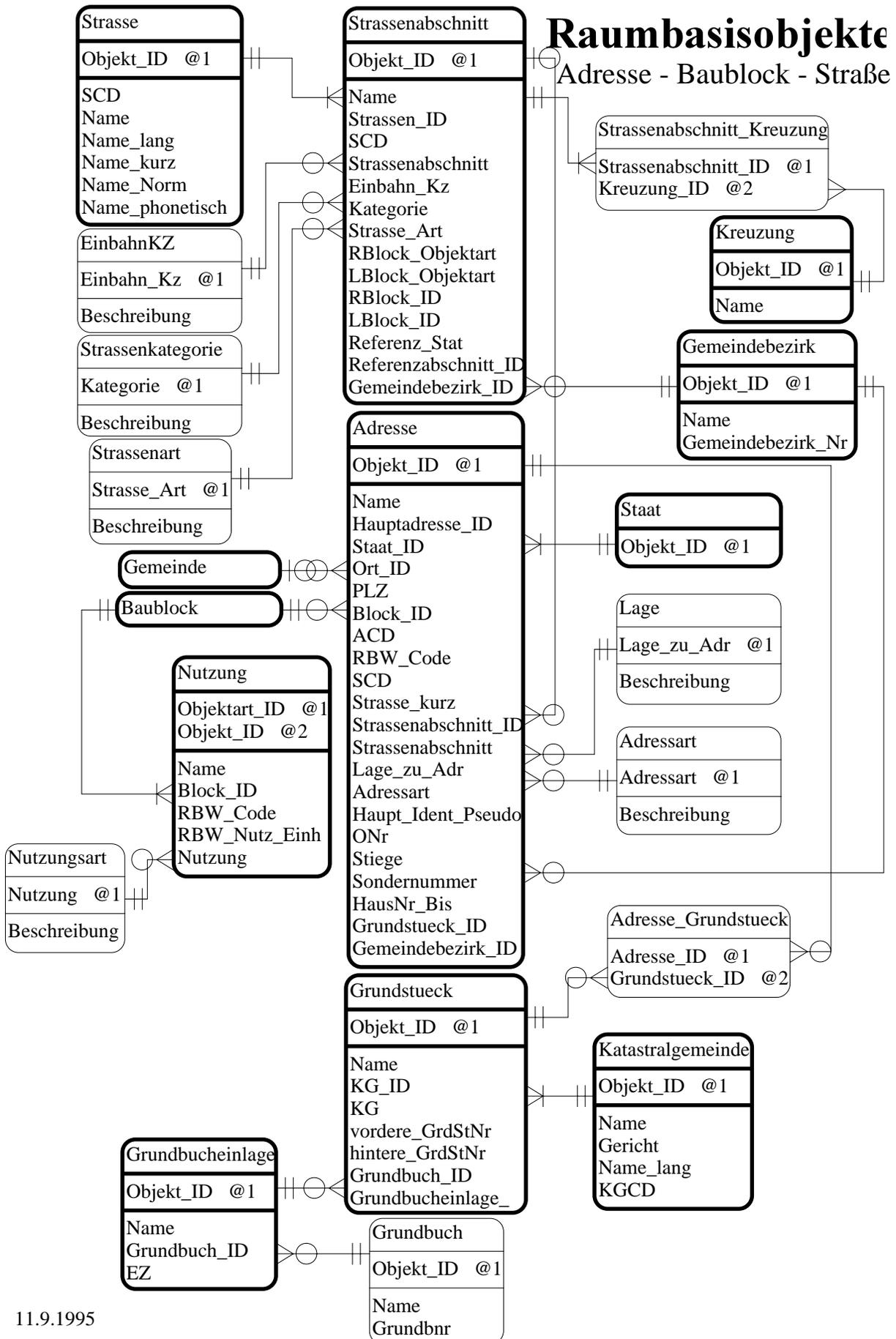
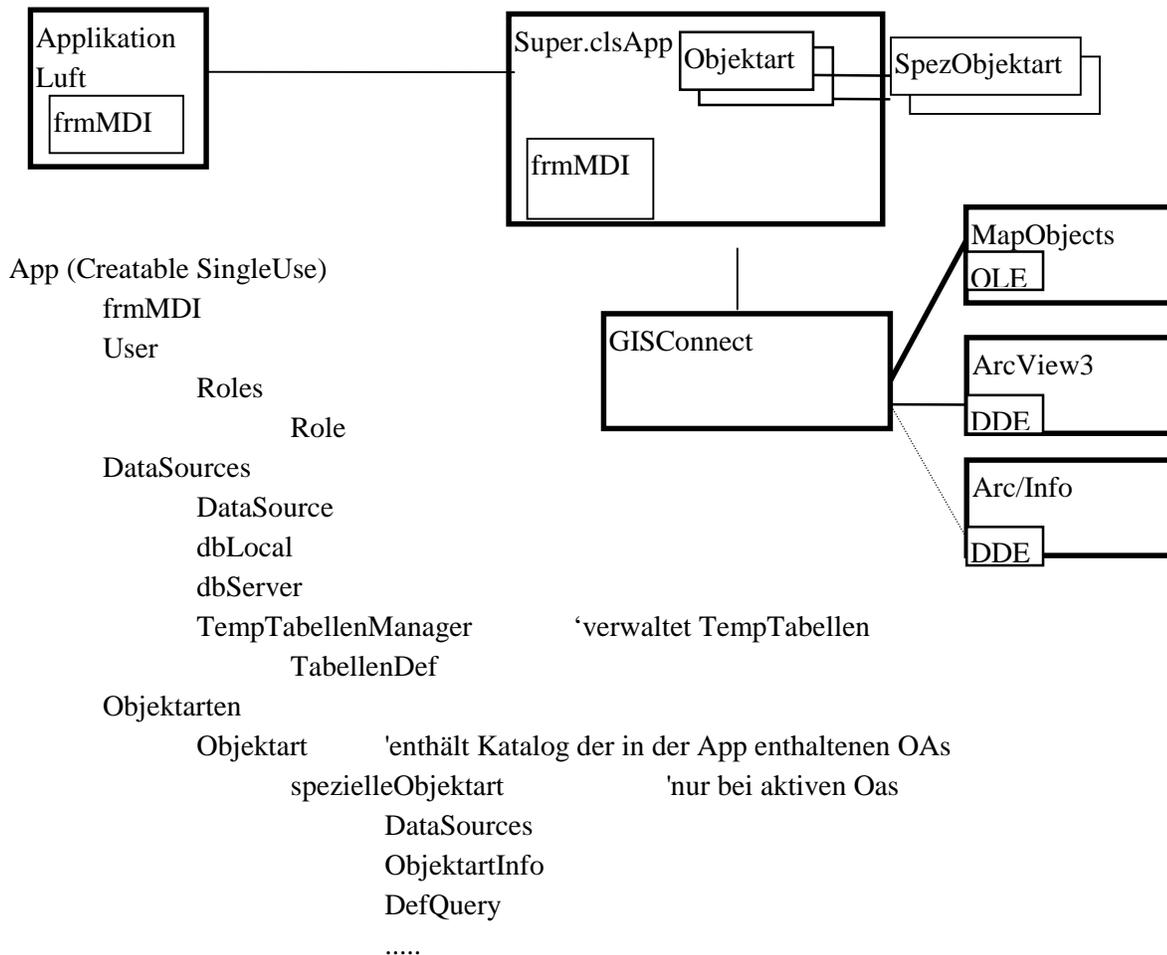


Abb 4: objektbezogenes Datenmodell - Beispiel

3.2. Applikationsmodellierung für OLE und GIS

Für alle Objekte ist festzulegen und zu beschreiben, welche Eigenschaften sie offenzulegen haben. Diese Beschreibung kann nur in einem iterativen Prozeß abgestimmt und definiert werden. Im Laufe der Entwicklung werden sich immer wieder neue Eigenschaftserfordernisse ergeben, die dazu verleiten, auch gekapselte Daten als Eigenschaften offenzulegen, was bei späteren Implementationsänderungen zu Problemen führt. Eine solche einfache Objektfunktionsbeschreibung könnte wie folgt aussehen:

3.2.1. Übersicht (Ausschnitt):



3.2.2. Detail (Ausschnitt):

clsGISView

'unabhängig vom aktiven GIS, dient der Kommunikation von clsGISConnect mit caller

'enthält alle vom Benutzer angesprochenen Views, max. Wien + Ostregion + Österreich + Europa

Public Property Get/Let sName As String

'Name der View, zugleich Key in der View-Collection, unique

Public Property Get/Let sHelpThema As String

'Name des HilfeThemas zur View

Public Property Get/Set GrenzenThema As clsGISThema

'Verweis auf das Hintergrundthema mit den Grenzen

.....

Public Property Themen As Collection

'of clsGISThemen

clsGISThema

‘reine Kommunikationsklasse

Public Property Get sName As String

‘Name des Themas, zugleich Key in der Themen-Collection der View, unique

.....

Public Property Get nDatasourceArt As Long

‘gisShapeFile, gisSDE, gisCoverage

Public Property Get sDatasource As String

‘Full Path to Shapefile, SDE-Dataset, Coveragename

Public Property Get sGeoServerName As String

‘Servername wird automatisch aus sDataSource extrahiert

Public Property Get sGeoDatabase As String

‘DataBase/Dir wird automatisch aus sDataSource extrahiert

.....

Public Function nInitialize(sName As String, sDataSurce As String, nDataSourceArt As Long, Optional sHelpThema As Variant) As Integer

‘zerlegt den SourceString in seine Einzelteile

4. CONCLUSIO

Das Paradigma der objektorientierten Datenverarbeitung wird mit den nun verfügbaren Werkzeugen im Entwicklungs- und GIS-Bereich zum erstenmal greifbar. Es wird aber noch einige Zeit brauchen, bis dieses Paradigma produktiv in Applikationen umgesetzt werden kann, und die daraus resultierenden Vorteile genutzt werden können. Nichtsdestotrotz führt kein Weg um dieses Konzept herum und gerade in allen Anwendungen, die sich mit Umwelttatbeständen auseinandersetzen müssen, oder die mit einem GIS zusammenarbeiten müssen, ist die möglichst frühzeitige Nutzung dieser neuen Technologie geboten.

5. LITERATUR:

- Kutsche, R.-D./Wasserroth S./Schöning C. (1996): Objektorientierte Methoden, ein adäquates Mittel zur Entwicklung von Umweltinformationssystemen , in [Lessing 1996]
- Lessing, H.(Hsg.)/Lipeck U. (1996): Informatik für den Umweltschutz, Hannover
- Luttermann H. (1996): Zur raum-zeitlichen Modellierung von Umweltdaten, in [Lessing 1996]
- Object Management Group (1995): What Is CORBA?, <http://www.omg.org/corba/corb2prf.htm>
- OpenGIS Consortium (1996): OpenGIS-Guide, <http://ogis.org/menu.html>
- Page, B./Häuslein A./Greve K. (1993): Das Hamburger Umweltinformationssystem (HUIS) - Aufgabenstellung und Konzeption, Hamburg
- Schöning C./Steinhau R./Wagener W. (1996): LUIS - Landesumweltinformationssystem Brandenburg, in [Lessing 1996]
- Van Nouhuys J./Jahr B. (1996): Entwicklung des Geographischen Informationssystems Umwelt - GISU - im Umweltbundesamt, in [Lessing 1996]