

# Entwicklung von EVU-Fachschalen unter Nutzung von UML\*

Albert Zimmermann

## Zusammenfassung

Seit ungefähr 25 Jahren werden in der Softwaretechnik Werkzeuge eingesetzt, die den Entwicklungsprozess komplexer Informationssysteme unterstützen. Mit der voranschreitenden Verbreitung objektorientierter Programmiersprachen hat sich auf diesem Gebiet die Modellierungssprache UML als Standard herausgebildet. Auch für die Entwicklung von GIS-Fachschalen, insbesondere im Bereich der Versorgungswirtschaft, bietet sich die Verwendung von UML zur Unterstützung verschiedener Entwicklungsphasen an. An drei ausgewählten Beispielen werden Möglichkeiten zur Verwendung von UML bei der Entwicklung von EVU-Fachschalen dargestellt.

## Summary

*For the last 25 years, software engineering tools have been used to support the development of complex information systems. As object orientated programming languages become more widespread, Unified Modelling Language (UML) has become accepted as the industry standard. UML is also appropriate for the development of GIS applications, particularly for public utilities. This paper gives three examples of the use of UML in the development of public utility supply networks.*

## 1 Einleitung

EVU-Fachschalen sind als spezielle Geoinformationssysteme heute in Energieversorgungsunternehmen wichtige Werkzeuge für Arbeiten zur Planung und Dokumentation der Versorgungsnetze. Während noch vor zehn Jahren die digital geführte Betriebsmitteldokumentation im Vordergrund der Aufgaben eines Netzinformationssystems stand, können heute – überwiegend nach Ende der Ersterfassungsphase – die vervollständigten Betriebsmitteldaten auch sinnvoll für komplexere betriebliche Aufgaben verwendet werden. Mit dieser Aufgabenverlagerung kann auch ein Wechsel zu einem neuen Geoinformationssystem notwendig werden. Die Anforderungen an ein zukünftiges Netzinformationssystem spiegeln dabei nicht nur spezielle technische Realisierungen der Betriebsmittel, sondern auch individuelle Organisationsstrukturen der Unternehmen und gestiegene Ansprüche

an den Datenaustausch mit anderen Informationssystemen (z. B. der kaufmännischen Abteilung) wieder. Ein Netzinformationssystem »von der Stange«, das als EVU-Fachschale von einem GIS-Hersteller angeboten wird, stellt dafür i. Allg. nur einen allgemeinen Rahmen zur Verfügung, der an die individuellen Umgebungsbedingungen eines Unternehmens angepasst werden muss. In selteneren Fällen beschließt ein Energieversorgungsunternehmen sogar, eine individuell angepasste Fachschale vollständig entwickeln zu lassen. In jedem Fall aber treten während des Umstellungsprozesses auf eine neue GIS-Plattform Phasen ein, in denen die Systemanforderungen angemessen dokumentiert werden müssen. An ausgewählten Beispielen werden die Möglichkeiten, die die Modellierungssprache UML zur Unterstützung der genannten Phasen bietet, dargestellt. Dabei soll auch auf GIS-spezifische Darstellungsmöglichkeiten der Sprache UML eingegangen werden.

## 2 Grundlagen der Software-Technik

### 2.1 Computerunterstützte Entwicklungswerkzeuge

Aus Sicht der Softwaretechnik stellt eine GIS-Fachschale ein spezielles Informationssystem dar, das neben den allgemein üblichen alphanumerischen Sachdaten auch über strukturierte grafische Daten verfügt. Insofern unterscheidet sich eine GIS-Fachschale bezüglich ihrer Komplexität nicht von gewöhnlichen Informationssystemen. Die Systemkomplexität steigt ihrerseits (Vetter 1990)

- mit der Anzahl unterschiedlicher Daten und den gegenseitigen Abhängigkeiten (Beziehungen),
- mit der Anzahl der Ereignisse, die Geschäftsprozesse auslösen, und den beteiligten Nutzern des Systems, sowie mit den speziellen betrieblichen Strukturen der Unternehmen,
- mit der Anzahl unterschiedlicher Zustände, die die Daten während ihrer Lebensdauer im System annehmen können.

In den 1970-er Jahren werden erste Versuche unternommen, die Komplexität von Informationssystemen mit geeigneten Modellen und Methoden zu erfassen. Diese greifen in alle Phasen des Software-Entwicklungsprozesses ein. Auf dieser Basis entstehen computerunterstützte Entwicklungswerkzeuge, die allgemein unter dem Sammelbegriff CASE (Computer Aided Software Engineering) bekannt werden. Ihre Aufgabe besteht darin, mit speziellen

\* Nach einem Vortrag gehalten im Rahmen der DVW-Fortbildungsveranstaltung Kommunale Geoinformationssysteme (KOMGIS), 21. Juni, Bonn 2001.

Editoren die zugrunde liegenden Modelle und Methoden abzubilden, aus den zusammengetragenen Anforderungen Dokumentationen zu erstellen, die festgelegten Spezifikationen auf Plausibilität zu prüfen und schließlich Quelltexte oder Quelltextgerüste für die Implementierung (Programmierung) zu generieren.

Bislang haben CASE-Werkzeuge bei der Entwicklung von GIS-Fachschalen nur eine untergeordnete Rolle gespielt. Ihre Verwendung beschränkt sich weitgehend auf die Dokumentation der Anforderungen und Spezifikationen für die zugeordneten Fachdaten. Beim Übergang von der Entwurfsphase zur Implementierung (Erstellung der Quelltexte) treten Systembrüche auf, die die weitere Verwendung in GIS-Projekten verhindern:

- Für GIS-spezifische Programmiersprachen sind i. Allg. keine geeigneten Sprachgeneratoren verfügbar.
- Die Funktionalitäten der CASE-Werkzeuge beschränken sich auf nahezu ausschließlich alphanumerische Anwendungsfälle, typische Problemfälle aus GIS-Anwendungen können in den CASE-Modellen nicht abgebildet werden oder führen zu Redundanzen mit bereits verfügbaren Funktionalitäten eines GIS (z. B. bei der Topologie).

Eine Ausnahme bildet das GIS der Fa. GE Smallworld, das bereits seit längerem über ein eigenes CASE-Werkzeug zur Entwicklung und Generierung von Datenbank-schemata für Anwendungen dieser speziellen GIS-Plattform verfügt.

## 2.2 Objektorientierte Systementwicklung

Mit dem Wechsel von textorientierten zu grafisch-interaktiven Benutzerschnittstellen haben in der Software-Technik auch die objektorientierten Programmiersprachen (Abk.: OOP) ihre prozeduralen Vorgänger weitgehend abgelöst. Die neuen OOP-Konzepte haben im Ergebnis dazu geführt, dass komplexe und graphisch interaktiv bedienbare Software-Produkte schneller und stabiler entwickelt werden können als mit den herkömmlichen prozeduralen Sprachen. Wesentliche Konzepte, die auch bei der objektorientierten Systementwicklung eine Rolle spielen, sind:

- **Klassen und Objekte**  
Fachlich zusammengehörende Daten können zu einem logischen Verbund, einem Objekt, zusammengefasst werden. Zu allen Objekten mit gleichen Eigenschaften gibt es einen Prototyp, nämlich eine Klasse. Die in einer Klasse zusammengefassten Daten werden als Attribute bezeichnet und beschreiben in ihrer Gesamtheit die Eigenschaften einer Klasse.
- **Datenkapselung und Methoden**  
Daten, die zu einem Objekt gehören, lassen sich gegenüber unkontrolliertem Zugriff schützen. Sie können nur mit solchen Funktionen abgefragt und verän-

dert werden, die eigens in der Klasse für diese Zwecke deklariert sind. Diese Funktionen sind ebenfalls Bestandteil einer Klassendeklaration und nur in Verbindung mit einem Objekt verwendbar. Die an Objekte gebundenen Funktionen werden als Methoden bezeichnet. Die Gesamtheit aller Methoden einer Klassen legt das dynamische Verhalten der Klasse bzw. der Objekte fest. In einem objektorientierten Programm wird der Ablauf durch gegenseitigen Methodenaufruf der beteiligten Objekte realisiert (Botschaftenprinzip).

- **Vererbung**

Durch den Mechanismus der Vererbung übernehmen neu geschaffene Klassen (abgeleitete Klassen) die Eigenschaften und das Verhalten von bereits bestehenden Klassen (Basisklassen). Die abgeleiteten Klassen können um weitere Attribute und Methoden ergänzt werden und Methoden, die sie von den Basisklassen geerbt haben, verändern. Dieses Prinzip ermöglicht es, grundlegende Software-Module für unterschiedliche Projekte zu verwenden (Wiederverwendbarkeit).

## 2.3 Die Modellierungssprache UML

Die Entwicklung der Modellierungssprache UML beginnt 1994 mit der Entscheidung führender Fachleute auf dem Gebiet der objektorientierten Systementwicklung (Boochs et al. 1999), ihre Entwicklungen gemeinsam fortzusetzen. Das Ergebnis ihrer Arbeit veröffentlichen sie 1997 als Unified Modelling Language (UML, Version 1.0); es wird noch im gleichen Jahr von der Objekt Management Group (OMG) als Standard akzeptiert und darf heute allgemein als Stand der Technik beim Software-Prozess für komplexe Informationssysteme gelten (Balzert 1999, Neumann 1998). Alle bedeutenden Hersteller von Entwicklungswerkzeugen verwenden mittlerweile Modelle und Methoden, die dem UML-Standard folgen.

Die Modellierungssprache UML verfügt über grafische (Symbole und Diagramme) und syntaktische (definierte Sprachsequenzen) Elemente, die einen objektorientierten Systementwurf abbilden. Die in UML verfügbaren Modelle, Diagramme und Elemente zur logischen Struktur eines Informationssystems werden im folgenden Kapitel anhand eines Beispiels eingehender dargestellt.

Der wesentliche Vorteil, der UML auch im Hinblick auf die Entwicklung von GIS-Fachschalen interessant erscheinen lässt, besteht in der Erweiterbarkeit ihrer Modellelemente. Darüber wird in den anschließenden Kapiteln berichtet.

## 3 Modellierung betrieblicher Abläufe

Anhand betrieblicher Abläufe in der Hausanschlussstelle (Kundencenter) eines Energieversorgungsunternehmens

(EVU) werden verschiedene UML-Modellelemente sowie deren wechselseitige Verzahnungen vorgestellt. Das angegebene Beispiel ist weitgehend generalisiert; die skizzierte Vorgehensweise stellt exemplarisch einen Ansatz dar, der immer dann empfehlenswert ist, wenn bei den zukünftigen Benutzern keine konkreten Vorstellungen über die Systemfunktionalität bestehen.

### 3.1 Geschäftsprozesse

Am Anfang der Analyse eines Informationssystems steht die Untersuchung repräsentativer Geschäftsprozesse (Balzert 1999). Dabei handelt es sich um zusammenhängende Tätigkeiten, die einen komplexen Vorgang mit einem vorgegebenen Ergebnis abschließen sollen. Die Bearbeitung eines Geschäftsprozesses kann längere Zeit andauern, die Mitwirkung mehrerer Mitarbeiter oder externer Personen erfordern und Tätigkeiten umfassen, die sowohl mit als auch ohne Informationssystem durchzuführen sind. Geschäftsprozesse werden in Anwendungsfall-Diagrammen erfasst. Diese beschreiben die Beziehungen zwischen Geschäftsprozessen, die die Unterstützung durch ein Informationssystem erfordern, und den beteiligten Einrichtungen oder Personen, den Akteuren.

#### Beispiel: Anwendungsfall-Diagramm zur Hausanschlussverlegung

Innerhalb der Aufgaben einer Hausanschlusssstelle soll der betriebliche Ablauf einer Hausanschlussverlegung verfolgt werden. Ausgelöst wird dieser durch einen Kundenantrag. Bereits bei der Aufnahme des Antrags werden im Informationssystem wichtige Daten im System gespeichert. Nach Prüfung weiterer technischer Einzelheiten löst der zuständige Sachbearbeiter einen weiteren Geschäftsprozess aus: der Auftrag zur Verlegung des Hausanschlusses an eine externe Firma. Die Status- oder Zustandsinformation, die über den Bearbeitungsfortschritt

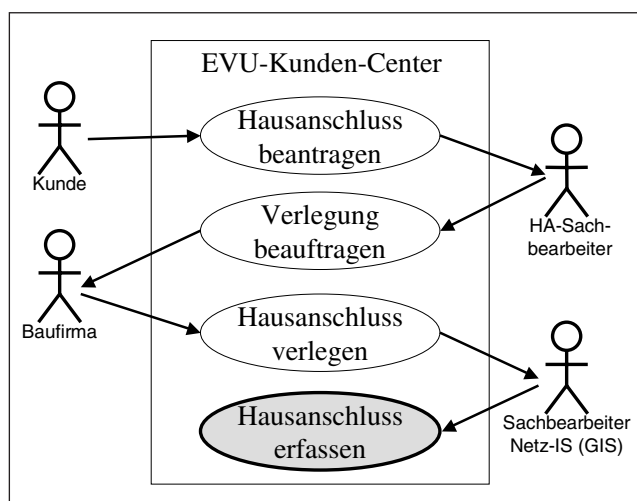


Abb. 1: Anwendungsfall-Diagramm für das Kundencenter eines EVU

des Kundenantrags Auskunft gibt, wird ebenfalls zusammen mit den Daten zum Bauauftrag im Informationssystem verwaltet. Darauf folgt die Hausanschlussverlegung, die mit der Fertigmeldung an das EVU endet. Dies löst einen weiteren Geschäftsprozess aus, an dem der Sachbearbeiter für die Betriebsmitteldokumentation (im Netzinformationssystem) als Akteur beteiligt ist; seine Aufgabe besteht darin, mit dem Leitungseinmessungsriss der Baufirma den neu verlegten Hausanschluss im GIS zu erfassen. Abbildung 1 gibt den beschriebenen Betriebsablauf in graphischer Form wieder.

### 3.2 Tätigkeitsabläufe

Die weiteren Komponenten der Systemkomplexität werden durch die Untersuchung von Tätigkeitsabläufen innerhalb der Geschäftsprozesse offengelegt. UML sieht für diesen Bearbeitungsschritt kein spezielles Modell oder Diagramm vor, jedoch können dafür Aktivitätsdiagramme verwendet werden (Balzert 1999). Aus den Bezeichnungen der Einzeltätigkeiten ergeben sich konkrete Angaben zu Daten bzw. Objekten und den darauf bezogenen Manipulationsfunktionen. Ferner können die Zustände der manipulierten Objekte erfasst werden. Am Ende des zweiten Analyseschrittes liegen konkrete Informationen über die erforderlichen Komponenten des Informationssystems vor. Dies betrifft:

- Objekte, deren Attribute und Manipulationsfunktionen (Methoden),
- Zustände, die während der Lebenszeit der Objekte auftreten,
- Beziehungen zwischen den Objekten.

#### Beispiel: Geschäftsprozess »Hausanschluss erfassen«

Die Bearbeitung des Geschäftsprozesses »Hausanschluss erfassen« setzt voraus, dass sich das betreffende Objekt »Hausanschluss« im Zustand »fertiggestellt« befindet. Wenn der Einmessungsriss der Beauftragten Firma vorliegt, kann der GIS-Sachbearbeiter mit den grafischen Erfassungsarbeiten beginnen. Zunächst muss er prüfen, ob die auf den Hausanschluss bezogenen Objekte, die Versorgungsleitung und das versorgte Gebäude, im Datenbestand existieren. Falls eines dieser Objekte fehlt, muss es vorab ebenfalls erfasst werden. Als Ergebnis dieses Analyseschrittes werden folgende Komponenten erhalten:

- die Zustände: »Hausanschlussleitung fertiggestellt«, »Hausanschlussleitung im Netzinformationssystem erfasst«;
- das Ereignis: »Einmessungsriss zum Hausanschluss wird vorgelegt«;
- die Objekte bzw. Klassen: »Versorgungsleitung«, »Gebäude«, »Hausanschluss«;
- die Beziehungen (Assoziationen): »Versorgungsleitung – Hausanschluss«, »Hausanschluss – Gebäude«;
- die Methoden: »erfassen« für Versorgungsleitung, Gebäude, Hausanschluss.

Das Aktivitätsdiagramm (Abbildung 2) zeigt die o. a. Komponenten in informeller syntaktischer Form.

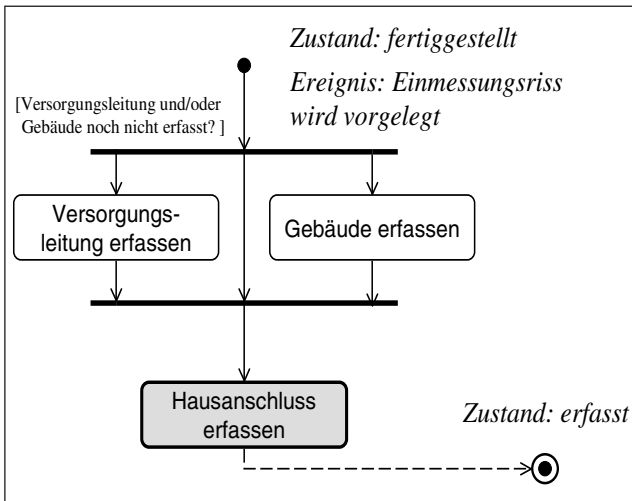


Abb. 2: Aktivitätsdiagramm für den Anwendungsfall »Hausanschluss erfassen«

### 3.3 Strukturierung der Systemkomponenten

Die Aufgabe der nachfolgenden Arbeitsschritte besteht darin, die festgestellten Komponenten des Systems in eine geordnete, sachgerechte Struktur zu bringen. Hierzu können verschiedene Diagrammtypen verwendet werden, deren Bearbeitungsreihenfolge frei wählbar ist. Diese bilden jeweils Teile der Komplexität des Systems ab. Erst die Gesamtheit der Diagramme ergibt eine vollständige Abbildung zur Struktur des Systems. Folgende Diagramme stehen zur Verfügung:

- In *Objektdiagrammen* werden Objekte und ihre Beziehungen untereinander während eines bestimmten Zeitraumes dargestellt. Der Zeitraum bezieht sich auf einen konkreten Zustand, den das Objekt während seiner Lebensdauer annehmen kann. Überwiegend handelt es sich um anonyme Objekte, die nur den Namen ihrer Klasse tragen. Attribute, die in der entsprechenden Phase von Bedeutung sind, sind aufgeführt.
- *Zustandsdiagramme* stellen den Lebenszyklus eines Objektes anhand der Zustände dar, die ein Objekt während seiner Existenz im System einnehmen kann. Der Lebenszyklus beginnt i. Allg. mit dem Erzeugen eines Objektes und endet mit dem Löschen aus dem Datenbestand. Diese Zustände sind durch abgerundete Rechtecke dargestellt und einzeln benannt. Damit ein Objekt seinen Zustand verändern kann, bedarf es i. Allg. einer Aktion, die in der Regel mit einem Methodenaufruf verbunden ist.
- Das dynamische Verhalten eines Informationssystems wird in *Sequenzdiagrammen* abgebildet. Diese zeigen gemäß dem Botschaftenprinzip die Kommunikation der Objekte untereinander, die durch Methodenaufrufe realisiert wird, und die in aller Regel durch die Aktivitäten der Benutzer (Akteure) ausgelöst werden.

**Beispiel: Komponenten und ihre strukturierte Darstellung**  
Das Ergebnis der Strukturierung der Komponenten zeigen die nachfolgenden Abbildungen.

Objekte (mit Attributen), Beziehungen und Zustände sind für einen begrenzten Ausschnitt im Objektdiagramm (Abbildung 3) dargestellt. Es zeigt den Hausanschluss im Zustand »erfasst«, mit seinen Beziehungen zu den Objekten Versorgungsleitung und Gebäude, sowie mit weiteren Attributen, die nach der Erfassung mit individuellen Werten belegt sind.

Die Herstellung der Beziehungen des Hausanschlusses zu seinen umgebenden Objekten macht weiterführende Aktionen erforderlich, die als Methoden abgebildet werden müssen. Wie aus dem Sequenzdiagramm (Abbildung 4) hervorgeht, werden diese Methoden ihrerseits in der Methode »Hausanschluss erfassen« verwendet und als Botschaften »zuordnen« auf die Versorgungsleitung und das Gebäude bezogen. Zuvor wird der Bearbeiter aufgefordert, mit der Methode »eingeben« die Fachdaten und die Grafik zum Hausanschluss zu erfassen.

Schließlich zeigt das Zustandsdiagramm (Abbildung 5) den Lebenszyklus des Hausanschlusses, von dem hier nur der Zustandsübergang von »Hausanschluss fertiggestellt« nach »Hausanschluss erfasst« näher untersucht wurde. In gleicher Weise müssen die anderen Zustandsübergänge bearbeitet werden.

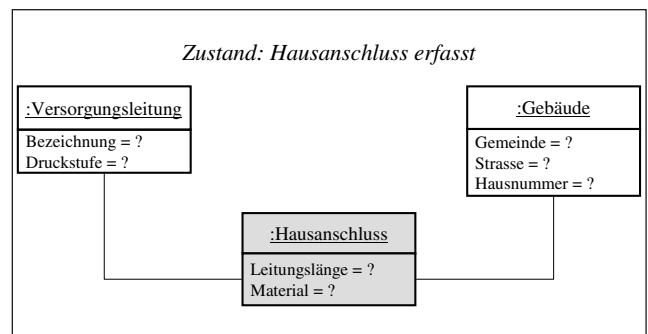


Abb. 3: Objektdiagramm für »Hausanschluss im Zustand erfasst«

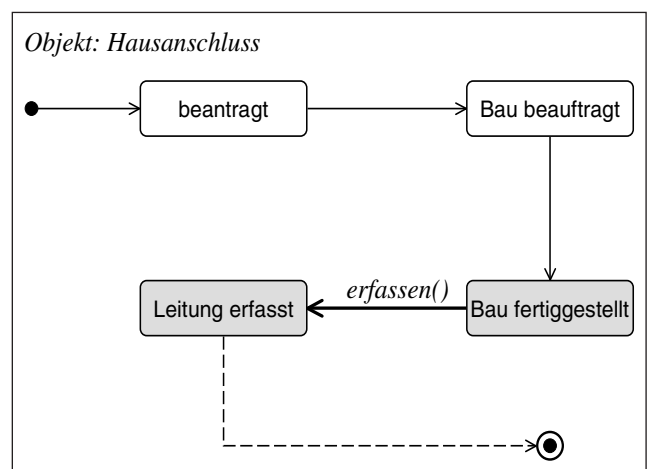


Abb. 4: Zustandsdiagramm für »Hausanschluss«



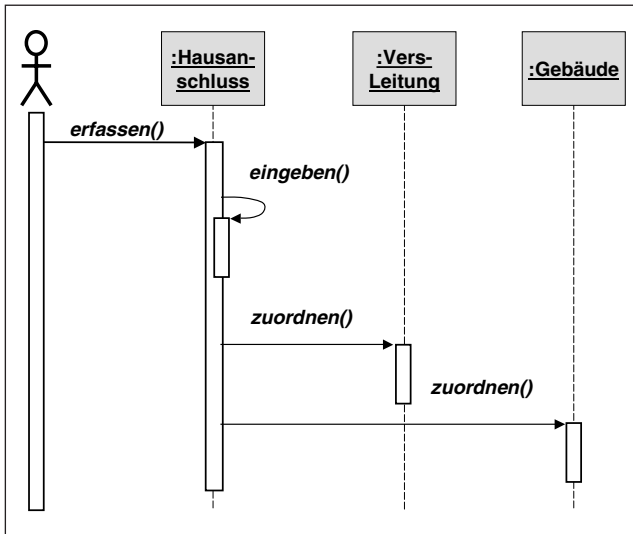


Abb. 5: Sequenzdiagramm zu »Hausanschluss erfassen«

### 3.4 Zusammenfassen der bearbeiteten Strukturen

Die Ergebnisse der o. a. Diagramme fließen schließlich in den Klassendiagrammen zusammen. Sie enthalten die Klassen mit allen Attributen und Methoden sowie die untereinander bestehenden Beziehungen. Die graphische Darstellung ähnelt einem Entity-Relationship-Diagramm, das von der Modellierung relationaler Datenbanksysteme her bekannt ist.

Abbildung 6 zeigt das Klassendiagramm für das behandelte Beispiel als Ergebnis einer Bearbeitung mit dem Entwicklungswerkzeug Rational Rose Enterprise Edition 2000 der Fa. Rational. Aus dem digital vorliegenden Modell lassen sich mit Hilfe weiterer Entwicklungswerkzeuge (hier: Datenbank- und Sprachgenerator OTRIS der Fa. Janus, Visual C++ und Access der Fa. Microsoft) lauffähige System-Prototypen entwickeln, die den zukünftigen Anwendern frühzeitig eine Vorstellung von der Erscheinungsform und der Funktionalität des späteren Informa-

tionssystems vermitteln können. Abbildung 7 zeigt einen Prototypen der Eingabemaske des vorangehenden Beispiels, der aus dem Klassenmodell generiert wurde.

## 4 Modellierung von Versorgungsnetzen

Im Verlauf der Entstehung und Nutzung von Netzinformationssystemen tritt mehrfach die Situation ein, dass Software-Komponenten abgelöst und gegen andere ausgetauscht werden müssen. Für eine langfristige strategische Planung erscheint es aber sinnvoll, wenn den Daten auch ein entsprechend langlebiges Modell zugrunde liegt, das einen Wechsel der GIS-Plattform übersteht. Aus heutiger Sicht können an solche Fachschalenmodelle folgende Anforderungen gestellt werden:

- Sie sollen systemneutral formuliert sein und keine herstellereinspezifischen Beschränkungen enthalten,
- die Strukturierung der Objekte soll netztopologischen Erfordernissen genügen, insbesondere Schaltsimulationen und Netzberechnungen unterstützen können und
- die Datenkommunikation mit kaufmännischen Systemen soll durch vereinbarte Schnittstellen konzeptionell vorbereitet sein.

Gegenwärtig bietet die Modellierungssprache UML, insbesondere das Klassendiagramm, die entsprechenden sprachlichen Möglichkeiten, langlebige Fachschalenmodelle zu formulieren. Gegenüber älteren Modellierungshilfsmitteln besitzt sie den Vorzug

- gezielt einsetzbarer Syntax für Beziehungen: Vererbung, Aggregation, Komposition und der Assoziation,
- der Integration benutzerdefinierter Datentypen und
- der Fähigkeit, nicht nur die Eigenschaften (Attribute) sondern auch das Verhalten (Methoden) von Objekten angemessen zu beschreiben.

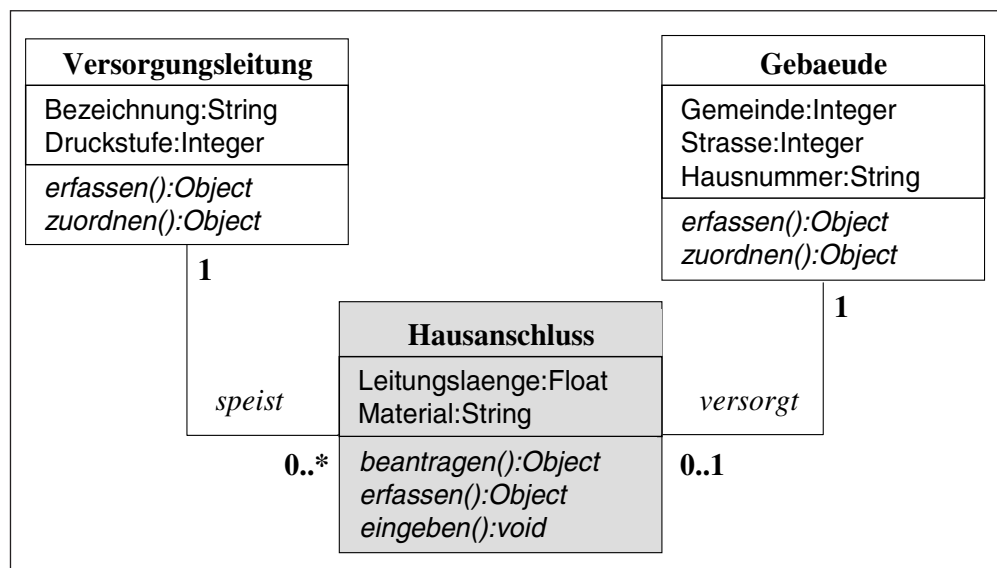


Abb. 6: Klassendiagramm für »Hausanschluss« und verwandte Klassen

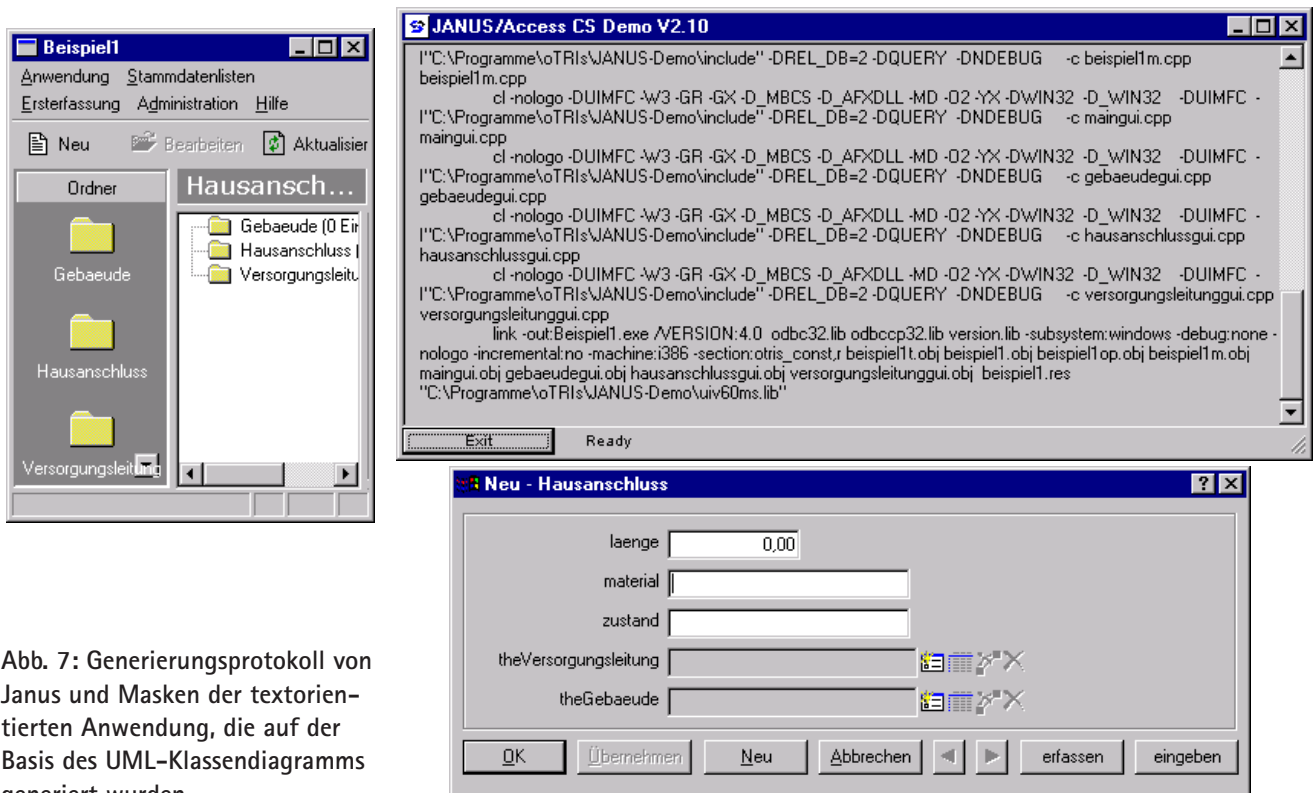


Abb. 7: Generierungsprotokoll von Janus und Masken der textorientierten Anwendung, die auf der Basis des UML-Klassendiagramms generiert wurden.

Am Beispiel eines Fachschalenmodells für das Wasserversorgungsnetz eines städtischen Versorgungsunternehmens, das im Rahmen eines gemeinsamen Projektes mit der Fachhochschule Neubrandenburg entwickelt wurde, sollen die sprachlichen Ausdrucksmöglichkeiten des UML-Modells eingehender dargestellt werden.

#### 4.1 Netztopologie und Schaltsimulation

Die grundlegende organisatorische Struktur eines Versorgungsnetzes geht von der Netztopologie aus, die ein Versorgungsnetz als geordnetes Gebilde von Netzkanten und Netzknoten abstrahiert. Netzkanten dienen zum Transport des Versorgungsmediums, während das Versorgungsmedium in Netzknoten verteilt, transformiert oder verbraucht wird. Auch müssen topologische Knoten dort gesetzt werden, wo sich die physikalischen Randbedingungen für den Transport verändern, oder das Versorgungsnetz endet. Eine weitere, spezielle Form von Knoten stellen Schaltstellen dar, an denen der Transport unterbrochen werden kann. Diese Knoten-Kanten-Struktur lässt sich in einem einfachen Beziehungsgraph abbilden, bei dem genau zwei Netzknoten eine Kante begrenzen und umgekehrt theoretisch unbegrenzt viele Kanten von einem Knoten abzweigen können.

Im Wasserversorgungsnetz finden sich mehrere Betriebsmitteltypen, die die Funktion topologischer Knoten besitzen (siehe Abbildung 9). Repräsentativ für ihre netztopologische Funktion steht die Basisklasse »Netzknoten«, von der die abgeleiteten Klassen »Uebergang«,

»Formstueck« und »Anlage« die Attribute und Methoden sowie die Beziehung zum Wasserstrang (topologische Kante) erben. Eine weitere Vererbungsstruktur ist mit der Klasse »Armatur« gegeben, die durch die konkreten Betriebsmittel »KKS« (Kathodischer Korrosionsschutz), »Entlueftung« und »Rueckschlagklappe« repräsentiert wird. Diese Betriebsmitteltypen sind mit den Wassersträngen verbunden ohne aber die Netztopologie zu beeinflussen.

Eine besondere Bedeutung in Wasserversorgungsnetzen besitzen die Schieber. Diese schaltbaren Betriebsmittel befinden sich häufig an Abgängen von Verzweigungsknoten und besitzen je nach ihrem Schaltzustand entweder Knoten- oder Kanteneigenschaften. In relationalen Datenbankentwürfen werden Schieber deshalb oft als topologische Knoten deklariert, was aber in der Praxis zu komplexen künstlichen Gebilden von eng beieinanderliegenden oder angrenzenden Knoten führt. UML verfügt mit der Beziehungsklasse über ein geeignetes Modellelement, mit dem das Schaltproblem gelöst werden kann. Beziehungsklassen können dazu verwendet werden, um ergänzende Eigenschaften und Verhaltensweisen an Beziehungen zu binden. Im Fall des Wasserversorgungsnetzes wird eine Klasse »Topologische Verbindung« deklariert, die als Träger der Information dient, ob zwischen einem Knoten und einer Kante eine permanente oder schaltbare Verbindung oder eine permanente Trennstelle vorliegt. Im Fall einer schaltbaren Verbindung muss mit der topologischen Verknüpfung ein Schieber in Beziehung stehen, der den aktuellen Schaltzustand angibt.

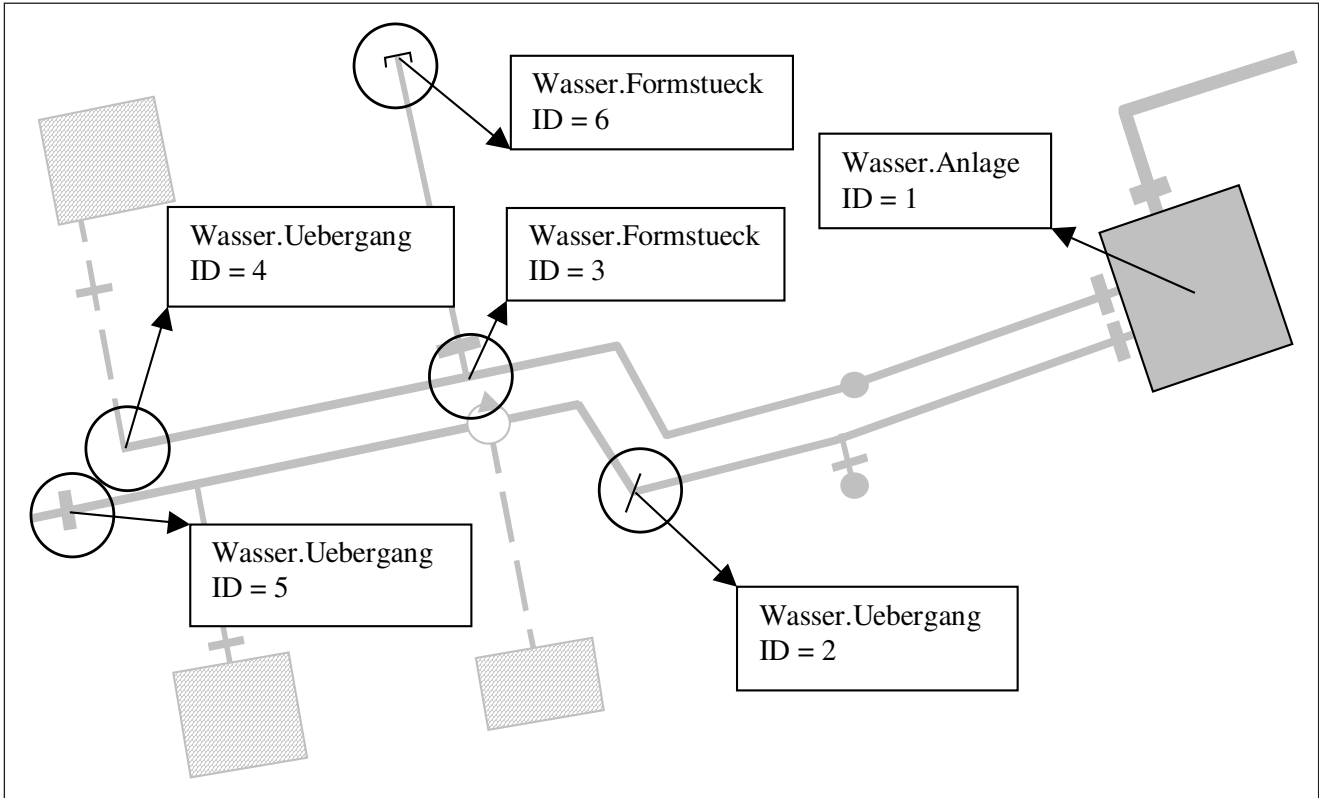


Abb. 8: Beispielausschnitt für ein Wasserversorgungsnetz mit Darstellung der topologischen Knoten

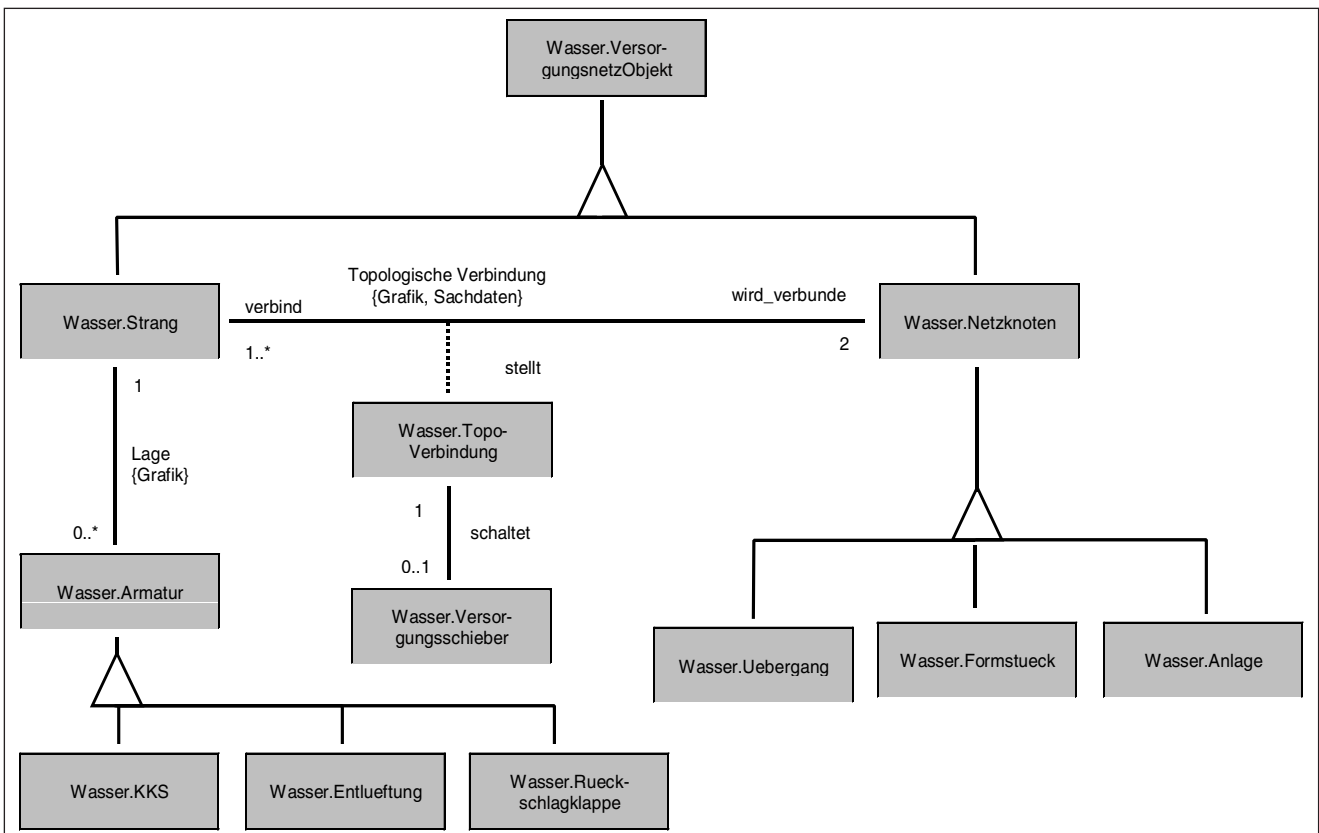


Abb. 9: Klassendiagramm zum Fachschalenmodell für das Wasserversorgungsnetz

## 4.2 Integration GIS-spezifischer Datentypen

Zur Abbildung GIS-spezifischer Objekte wie Punkte, Linien und Flächen sowie Texte und Raster verfügt UML über die Möglichkeit der Integration anwendungsorientierter Datentypen. Aus Sicht der objektorientierten Programmierung handelt es sich dabei um Klassen mit eigenen Eigenschaften und Verhaltensweisen, die ihrerseits als Datentypen für Attribute in komplexeren Klassen verwendet werden können. In einer simplen Modellierungsvariante stehen die GIS-bezogenen Objekttypen nur als Referenzen für die Klassen, die von den verschiedenen GIS-Plattformen zur Modellierung der Fachanwendungen zur Verfügung gestellt werden. An die graphischen GIS-Komponenten werden in diesem Fall dabei außer den grundlegenden Erfassungs- und Manipulationsfunktionen keine weiteren funktionalen Anforderungen gestellt.

Das Klassendiagramm in Abbildung 10 zeigt beispielhaft eine Realisierung der o. g. Entwurfsvariante. Die grafischen Komponenten der verschiedenen Betriebsmittel werden als Attribute geführt. Wegen der unterschiedlichen Darstellung der Betriebsmittelgrafik in verschiedenen Planwerken sind i. Allg. mehrere graphische Attribute zu führen; diese lassen sich in ihren Attributnamen durch angehängte Maßstabszahlen unterscheiden.

Ein Verifikation des vorgestellten Fachschalenmodellentwurfs wurde auf Basis von Autocad Map 2000 und Microsoft Access durchgeführt. Die Übertragung des objektorientierten Klassenmodells auf die Datenbankstruktur von Access erfordert eine objektrationale Abbildung, von der insbesondere die Vererbungsstrukturen und die Beziehungsklassen betroffen sind (Balzert 2000, Börries 1998). Im Ergebnis konnte nachgewiesen werden, dass das entworfene Fachschalenmodell in die Strukturen einfacher Erfassungswerkzeuge abgebildet werden kann.

## 5 GIS-spezifische Modellerweiterungen in UML

Wie das vorangegangene Beispiel zeigt, lassen sich GIS-Modellelemente formal in Klassendiagrammen integrieren. Für Arbeiten während der Analysephase und zur Dokumentation von Systemanforderungen ist dies auch ausreichend. Soll aber der gesamte Entwicklungsprozess mit Unterstützung von CASE-Werkzeugen durchgeführt werden, sind zusätzliche GIS-spezifische Modellerweiterungen, angepasste Werkzeuge und schließlich auch geeignete Generatoren für GIS-spezifische Sprachen erforder-

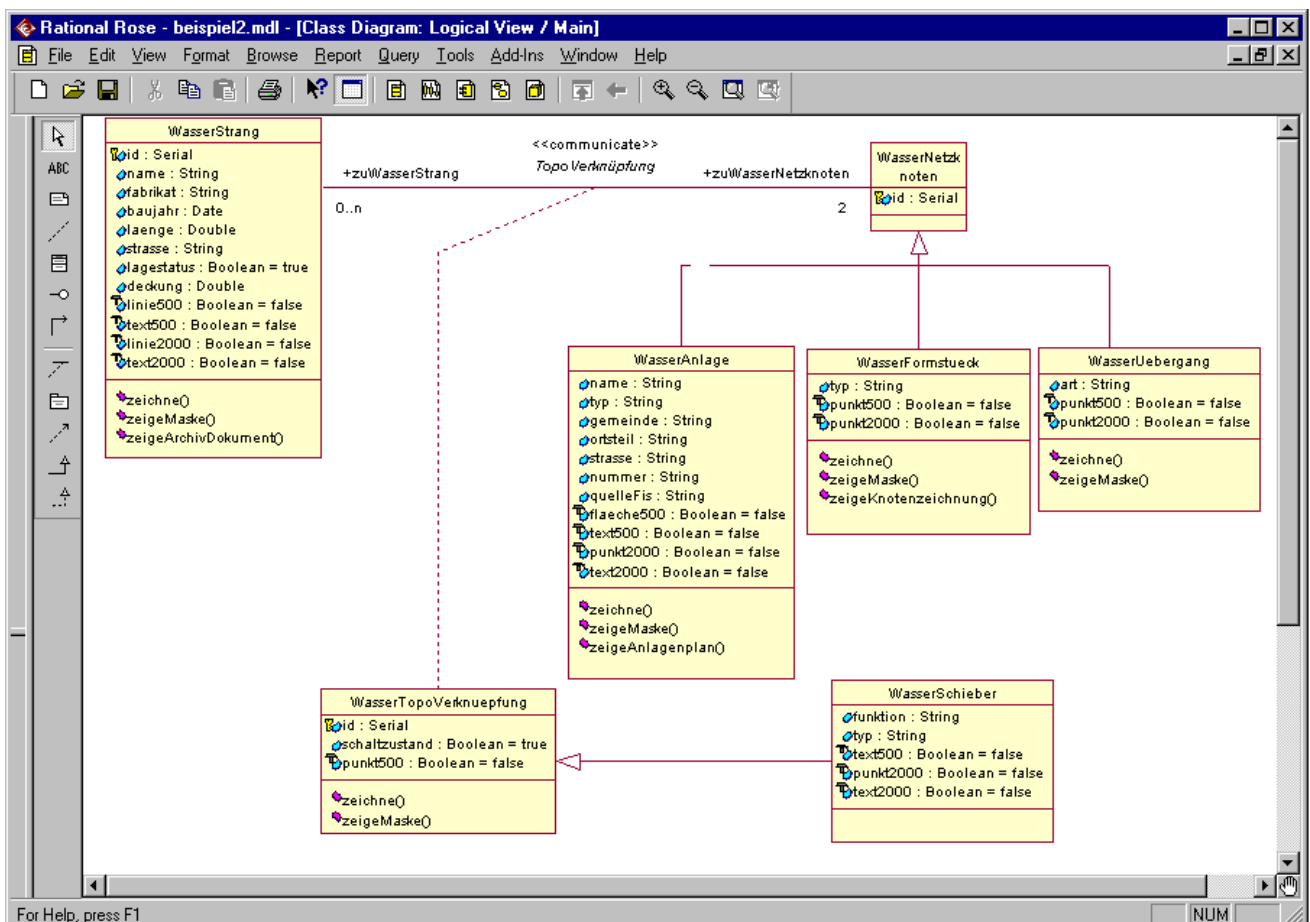


Abb. 10: Klassendiagramm zum Fachschalenmodell für das Wasserversorgungsnetz mit ergänzten Attributen und Methoden



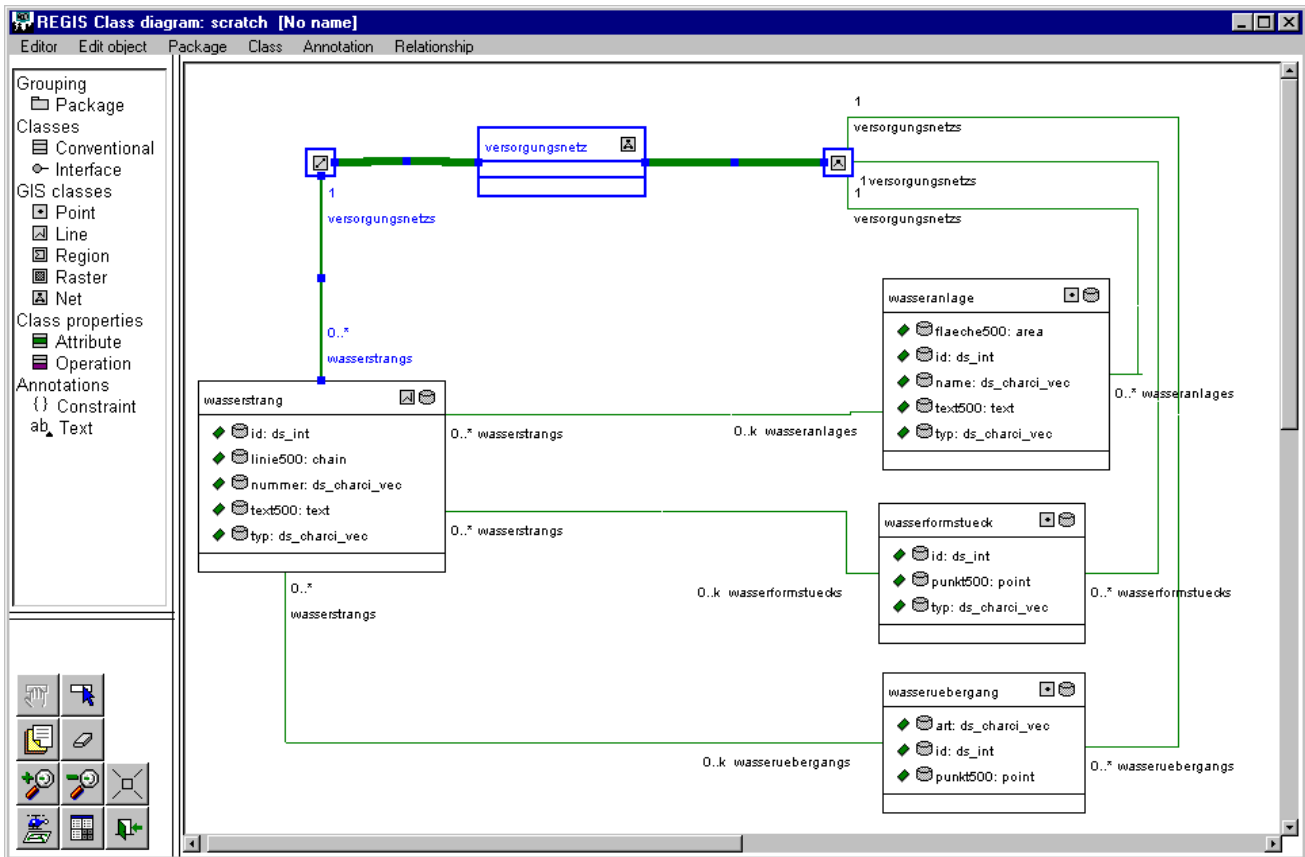


Abb. 11: Klassendiagramm zum Fachschalenmodell für das Wasserversorgungsnetz mit GIS-spezifischen Modellelementen von REGIS

derlich. Ein entsprechendes CASE-Werkzeug, das diese GIS-spezifischen Anforderungen auf der Basis von UML aufgreift und in weiten Teilen realisiert, wird nachfolgend vorgestellt.

### 5.1 Modellerweiterung durch Stereotypen

Eine der Erweiterungsmöglichkeiten von UML sind Stereotypen. Es handelt sich dabei um besondere Klassen mit spezialisierten Eigenschaften und Verhaltensweisen. UML verwendet diese Form der Erweiterung selbst für weitere vordefinierte Modellelemente, die formaler Bestandteil der Sprache sind [Neumann, 1998]. UML-basierte CASE-Werkzeuge verfügen i. Allg. über eigene Stereotypen, deren Eigenschaften und Verhalten in den Werkzeugen implementiert sind.

### 5.2 GIS-spezifische Stereotypen in REGIS

Ein UML-basiertes CASE-Werkzeug, das die speziellen Anforderungen von GIS-Anwendungen aufgreift, ist das System REGIS der Fa. Isoware, Hagen. REGIS verfügt über spezielle GIS-Klassen, die als Stereotypen implementiert sind. Dazu gehören die den geometrischen Ausprägungen (sogenannte Hauptgeometrien) entsprechen-

den Typen Point, Line, Region und Raster. In Klassendiagrammen werden sie wie gewöhnliche Klassen mit Attributen und Methoden dargestellt. Erkennbar ist der Unterschied nur anhand eines speziellen Symbols (engl.: icon), das dem Stereotyp zugewiesen ist. Zusätzlich lassen sich den GIS-Objekten weitere geometrische Attribute zuweisen, die jedoch begrenzte geometrische Funktionalität besitzen. Auch bei den Klassenbeziehungen verfügt REGIS über weitere Ergänzungen, die für geometrische bzw. topologische Auswertungen zwischen GIS-Objekten von Bedeutung sind.

Das Klassendiagramm in Abbildung 11 greift das Modell aus dem vorangegangenen Kapitel wieder auf. Die Knoten-Betriebsmittel »Wasseranlage«, »Wasserformstueck« und »Wasserübergang« sind in diesem Klassendiagramm als spezielle GIS-Klassen vom Typ »Point« ausgebildet. Die weiteren graphischen Eigenschaften werden lediglich als geometrische Attribute geführt. Entsprechend ihrer topologischen Eigenschaft werden die Betriebsmittel vom Typ »Wasserstrang« als »Line« geführt.

Von besonderer Bedeutung für die Modellierung von Versorgungsnetzen ist ein komplexes Konstrukt, das im Klassendiagramm (Abbildung 11) in blau und grün hervorgehoben dargestellt ist und die Bezeichnung »Versorgungsnetz« trägt. Dabei handelt es sich ebenfalls um eine spezielle GIS-Klasse, die Klasse »Net«. Dieser Stereotyp

besteht aus zwei miteinander in Beziehung stehenden Verknüpfungspunkten, die als Basisklassen für topologische Kanten (linker Arm) und für topologische Knoten (rechter Arm) dienen. Wird eine Klasse mit einem dieser Verknüpfungssymbole verbunden, so wird sie damit als topologischer Knoten oder als Kante deklariert und übernimmt automatisch die dadurch festgelegten Eigenschaften und Verhaltensweisen.

Bezüglich der weiterführenden Entwurfsphasen setzt REGIS einen deutlichen Schwerpunkt auf das Geoinformationssystem der Fa. GE Smallworld und die Programmiersprache MAGIK, in der Smallworld GIS implementiert ist. Dies zeigt sich u. a. darin, dass sich ein Teil der Eingabemasken ausschließlich auf die Modellstrukturen von Smallworld GIS bezieht.

## 6 Ausblick

Der im letzten Beispiel gezeigte Ansatz, bei dem die Modellierungssprache UML durch ihre Erweiterungsmöglichkeiten auch unmittelbar zur Lösung GIS-spezifischer Problemstellungen einsetzbar wird, kann nicht nur für MAGIK, sondern prinzipiell auch für andere objektorientierte Programmiersprachen nutzbar gemacht werden. Dafür ist eine geeignete Klassenbibliothek erforderlich, die die GIS-spezifischen UML-Erweiterungen abbildet. Aus heutiger Sicht würde – auch im Hinblick auf die Internet-Fähigkeit – die Bereitstellung eines Software-Moduls in Java sinnvoll sein.

Eine weitere Entwicklung ist aus Sicht des Autors bemerkenswert. Das Open GIS Consortium (OGC), eine Vereinigung bedeutender Entwickler von Geoinformationssystemen, hat sich zum Ziel gesetzt, ihre Produkte untereinander operabel zu machen. Diese Bestrebungen werden dazu führen, dass die heute noch überwiegend monolithischen Strukturen zahlreicher GIS-Produkte zugunsten einer Komponentenstruktur aufgebrochen werden, durch die Fachschalen aus Komponenten verschiedener Hersteller zusammengestellt werden können. UML könnte dabei als Modellierungssprache für die Zusammenstellung der GIS-Komponenten zu einer Fachschale verwendet werden. Schließlich setzt auch OGC zur Mo-

dellierung ihrer Standards die Sprache UML ein. Es bleibt abzuwarten, inwieweit die Herstellerfirmen, die an der Standardisierungsarbeit des OGC beteiligt sind, ihre Bemühungen auch diesbezüglich weiter vorantreiben.

### Anmerkungen, Danksagung

Verschiedene Abbildungen dieses Beitrags entstanden durch den Einsatz von CASE-Werkzeugen zu Testzwecken (Rational Rose 2000e Enterprise Edition der Fa. Rational, Janus der Fa. oTRIs, Visual C++ der Fa. Microsoft). Diese Produkte sind als Autorenlizenzen auf der CD-ROM zum Lehrbuch (Balzert, 2000) verfügbar. Der Fa. ISOWARE GmbH, Hagen, sei an dieser Stelle für die freundliche Überlassung einer Testlizenz des CASE-Werkzeugs REGIS herzlich gedankt.

### Literatur

- Balzert, H.: Lehrbuch der Objektmodellierung. Spektrum Akademischer Verlag, Heidelberg-Berlin, 1999
- Balzert, H.: Objektmodellierung in 7 Tagen. Spektrum Akademischer Verlag, Heidelberg-Berlin, 2000
- Behr, F.-J.: Strategisches GIS-Management, Grundlagen und Schritte zur System Einführung. Herbert Wichmann Verlag, Heidelberg, 1998
- Boochs, G.: Rumbaugh, J.: Jacobson, I.: Das UML-Benutzerhandbuch. Addison-Wesley, Bonn, 1999
- Börries, L.: Objektrelationale Transformation. In: DEHNHARDT, W., Anwendungsprogrammierung mit JDBC: 209–215, Carl Hanser Verlag, München-Wien, 1998
- Müller, F.: Was der Fall ist, Entwicklungswerkzeuge im Vergleich: OEW und OTW. IX-Magazin 2/2000:68–72, Verlag Heinz Heise, Hannover 2000
- Müller, F.: Der zweite Fall, Entwicklungswerkzeuge im Vergleich: ObjectiF und Rose. IX-Magazin 3/2000:80–83, Verlag Heinz Heise, Hannover 2000
- Neumann, H. A.: Objektorientierte Softwareentwicklung mit der Unified Modelling Language (UML). Carl Hanser Verlag, München-Wien, 1998
- Vetter, M.: Strategie der Anwendungssoftware-Entwicklung – Planung, Prinzipien, Konzepte, B. G. Teubner Verlag, Stuttgart, 1990

### Anschrift des Autors

Prof. Dr.-Ing. Albert Zimmermann  
 c/o Fachhochschule Bochum  
 FB Vermessungswesen und Geoinformatik  
 Lennershofstr. 140  
 44801 Bochum