

REGELORIENTIERTE ERZEUGUNG VON KARTEN-ENTWURFEN  
AUF GEOWISSENSCHAFTLICHEN DATENBANKEN

Michael Drawin, Karl Neumann, Hans-Dieter Ehrich

Informatik, Abt. Datenbanken  
Technische Universität Braunschweig  
Postfach 3329, D-3300 Braunschweig

**KURZFASSUNG:** In der vorliegenden Arbeit werden Sprachmittel zur Definition von Regeln zur Kartenerzeugung und deren Umsetzung auf die Benutzerschnittstelle eines geowissenschaftlichen Datenbanksystems vorgestellt. Diese Regeln legen die Darstellung von Klassen geowissenschaftlicher Objekte in ganzen Kartenarten fest. Der Entwurf einer konkreten Karte kann dann weitgehend automatisch, regelgesteuert durchgeführt werden.

**SCHLÜSSELWORTE:** geowissenschaftliches Datenbanksystem, Definition von Regeln, Kartenentwurf

**ABSTRACT:** This paper presents a language suitable to define rules representing a subset of cartographic knowledge. Also, the translation of these rules into the language of a geoscientific database system is sketched. Applying the rules to the process of map prototyping means to handle it almost automatically.

**KEYWORDS:** geoscientific database system, definition of rules, map prototyping

## 1. EINLEITUNG

Innerhalb des Schwerpunktprogrammes "Digitale Geowissenschaftliche Kartenwerke", in dem ein integriertes Konzept zur Handhabung von geowissenschaftlichen Datenbeständen und deren Umsetzung in Kartenwerke angestrebt wird /Vi85/, wurde eine Benutzerschnittstelle für ein entsprechendes Datenbanksystem vorgeschlagen. Die zugehörige Datenbanksprache, die sich an der relationalen Sprache QUEL /SWKH76/ orientiert, bietet unter anderem Definitions- und Manipulationsoperationen für Objekte mit geometrischen Datentypen sowie Anweisungen, die der Generierung von Karten dienen /RNLE85, LN86/. Ein erster Datenbankprototyp, der Teile dieser Datenbanksprache realisiert, befindet sich zur Zeit in der Testphase.

Es wurden deshalb erste Überlegungen angestellt, dieses geowissenschaftliche Datenbanksystem auch für den Aufbau eines Informationssystems innerhalb des Sonderforschungsbereiches "Wasser- und Stoffdynamik von Agrar-Ökosystemen" /Ro85/ zu nutzen. Dabei zeigte sich, daß für den geplanten Einsatz, das Erstellen von einfachen, aktuellen Karten, die Versuchsergebnisse aus dem Testgelände grob darstellen sollen, die existierende Benutzerschnittstelle zwar prinzipiell ausreicht, aber teilweise zu unbequem langen Anweisungsfolgen führt.

---

Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft gefördert (Az. SFB 179 B4 und Eh 75/3-2).

In der vorliegenden Arbeit wollen wir deshalb die Benutzerschnittstelle im Bereich der Kartenerstellung um Konzepte erweitern, die das bisherige Abbilden von "Geoobjekten" auf "Kartenobjekte" und deren Einfügen in gleichzeitig zu qualifizierende Karten vereinfachen. Mit Hilfe der neu einzuführenden Sprachmittel können oft benutzte Darstellungsstandards der Kartenerstellung ausgedrückt und beim Entwurf von Karten benutzt werden, wie z.B. die Tatsache, daß in vielen Karten die Namen von Städten abhängig von ihrer Einwohneranzahl in verschiedenen großen Schriften dargestellt werden.

Als Ausgangspunkt wird dazu im nächsten Abschnitt die Datenbanksprache des Geodatenbanksystems kurz vorgestellt. In Abschnitt 3 werden Grundregeln für die Kartenerzeugung diskutiert, der Art: "Welche Objektklassen werden in welchen Karten wie dargestellt". Entsprechende Sprachmittel zur Definition von Regeln und Klassen werden eingeführt und anhand von Beispielen erläutert. In Abschnitt 4 wird die neue Anweisung zur regelgestützten Kartenkonstruktion vorgestellt, während Abschnitt 5 auf die automatische Übersetzung dieser Anweisung in die ursprünglichen Kommandos der Datenbanksprache eingeht. Im letzten Abschnitt wird der aktuelle Stand der Realisierung des Gesamtsystems dargestellt, außerdem werden hier Möglichkeiten zur Optimierung aufgezeigt.

## 2. GEOOBJEKT-MODELL UND DATENBANKSPRACHE

In diesem Abschnitt wird das Geoobjekt-Modell und die zugehörige Datenbanksprache anhand von Beispielen sehr knapp vorgestellt. Für eine detailliertere Darstellung, insbesondere des Objekt-Modells wird auf /LN86/ verwiesen; /RNLE85/ enthält zusätzlich die Sprachsyntax.

Das für geowissenschaftliche Anwendungen entwickelte Objektmodell, kurz: Geoobjekt-Modell, stellt eine Erweiterung und Spezialisierung des bekannten Entity-Relationship-Modells (ER-Modell) /Ch76/ dar. Die Erweiterungen betreffen:

Geometrische Datentypen: es werden die Nicht-Standard-Datentypen "Punkte", "Linien" und "Flächen" mit zugehörigen Operationen wie Flächenschnitte, Längenberechnung von Linien usw. vorgesehen.

Komplexe Objekte: neben den atomaren Objektklassen können auch komplexe Objektklassen definiert werden, diese können Listen, Mengen oder Aggregationen von Subobjekten enthalten.

Generalisierungen: mehrere (Unter-)Objektklassen können zu Oberklassen generalisiert werden. Es ist möglich, daß diese Oberklassen wieder Unterklassen darstellen, so daß Generalisierungshierarchien entstehen.

In Abb.1 ist ein erweitertes ER-Schema dargestellt, das die aufgeführten Konstrukte enthält. Die beiden atomaren Objektklassen "SO<sub>2</sub>-Messung" und "Waldschäden" werden zur Oberklasse "Erhebungsstelle" generalisiert. "Regierungsbezirke" werden als komplexe Objekte modelliert, die jeweils eine Liste von Landkreisen enthalten. Sämtliche Objektklassen weisen ein Attribut "Geometrie" auf; so wird etwa die flächenhafte Ausdehnung von Städten als Menge von Polygonen dargestellt.

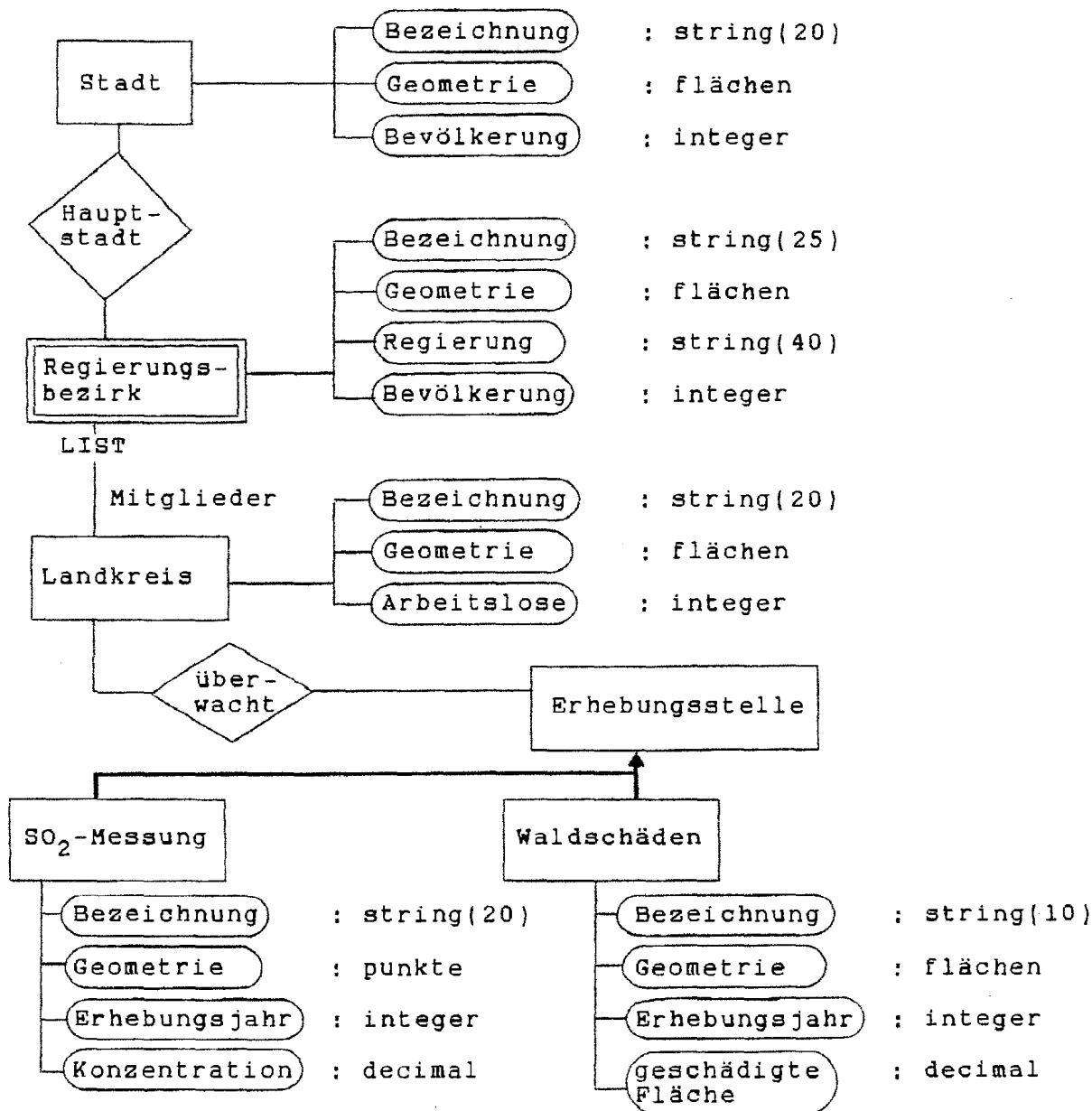


Abb.1: Beispiel eines erweiterten ER-Schemas

Zur Berücksichtigung der besonderen geowissenschaftlichen Anforderungen wird das skizzierte erweiterte ER-Modell zum Geobjekt-Modell spezialisiert, indem ein allen Geodatenbanken gemeinsames Teil-Schema vordefiniert wird und gewisse Regeln angegeben werden, wie der jeweils verbleibende Teil des Schemas aufgebaut werden muß. Dafür bietet das Geobjekt-Modell je zwei Hauptkonzepte Geobjekt und Kartenobjekt an.

Geobjekte stellen geowissenschaftliche Objekte dar, die in der Realität oder in der Vorstellungswelt existieren, z.B. Städte, Vegetationszonen, Bruchkanten usw. Dagegen sind Kartenobjekte die graphischen Repräsentationen von Geobjekten in (Land-)Karten. Während die Geobjekt-Klassen vom Benutzer relativ frei definiert werden können, bilden die sechs verschiedenen Kartenobjekt-Klassen und die Klasse der Karten den festen Teil des Schemas.

Abb.2 verdeutlicht diese Schema-Zweiteilung und gibt eine Übersicht über die wichtigsten Datenbankoperationen mit deren Zuordnung zu den beiden Schemateilen.

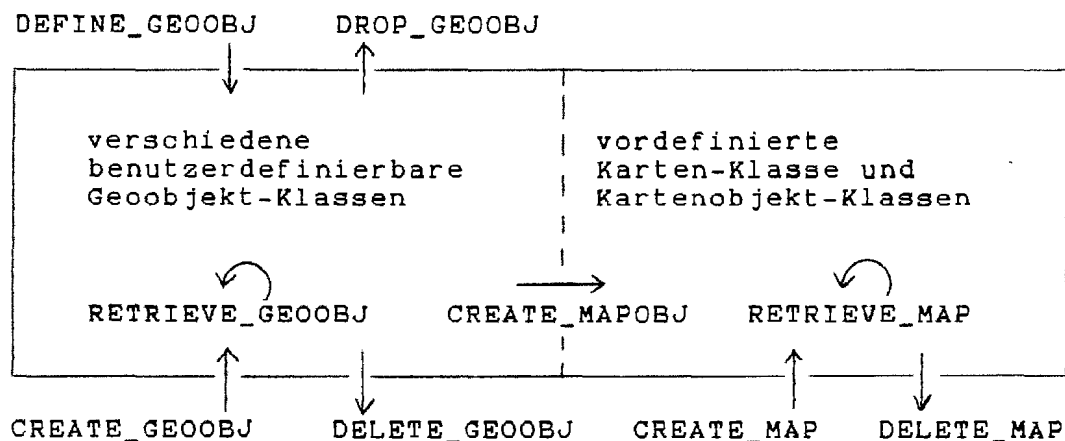


Abb.2: zweigeteiltes Schema mit Operationen

Eine atomare Geoobjekt-Klasse wird durch das Kommando "DEFINE\_GEOOBJ" definiert. Bei der Definition müssen wie üblich ein Klassenbezeichner sowie geeignete Attributnamen und Datentypen angegeben werden. Das Attribut "BEZEICHNUNG" ist dabei obligat, wogegen das "GEOMETRIE"-Attribut sowie beliebige weitere inhaltliche Attribute optional sind, z.B.:

```
DEFINE_GEOOBJ Landkreis
  (BEZEICHNUNG STRING(20),
   GEOMETRIE FLACHEN,
   Arbeitslose INTEGER);
```

Bei der Definition von komplexen Objektklassen muß zusätzlich die Art deren Zusammensetzung aus Subobjekten angegeben werden:

```
DEFINE_COMPOBJ Regierungsbezirk
  (BEZEICHNUNG STRING(25),
   GEOMETRIE FLACHEN,
   Regierung STRING(20),
   Bevölkerung INTEGER,
   Mitglieder LIST_OF Landkreis);
```

Klassen von Beziehungen, z.B. "Hauptstadt" zwischen den Objektklassen "Stadt" und "Regierungsbezirk", sowie Generalisierungshierarchien werden auf ähnliche Weise definiert.

Nachdem ein Datenbankschema definiert wurde, kann der Benutzer mit der Anweisung "CREATE\_GEOOBJ" oder per Massendatenlader (bulk-loading) Objektexemplare einfügen. Im folgenden Beispiel werden nur einige Zeichenketten-Konstanten angegeben; die Punkte stehen für weitere fehlende Konstanten.

```
CREATE_GEOOBJ Landkreis('Celle',{...},...);
CREATE_COMPOBJ Regierungsbezirk('Lüneburg',{...},...);
```

Das Wiederfinden, Aktualisieren und Löschen von Objekten stützt sich auf das Konzept der typisierten Objektvariablen, die mittels der "RANGE\_OF"-Anweisung deklariert werden müssen. Bei dieser Deklaration kann für Objektklassen, die ein Geometrie-Attribut aufweisen, ein Fenster in Weltkoordinaten angegeben werden (IN\_WINDOW). Die betreffende Variable überstreicht dann nur noch solche Objektexemplare, deren Geometrie innerhalb des angegebenen Fensters liegt.

Subobjekte werden durch eine spezielle Form der "UPDATE"-Anweisung zu komplexen Objekten hinzugefügt. Im nächsten Beispiel wird der Landkreis "Celle" dem Regierungsbezirk "Lüneburg" hinzugefügt. Es wird dabei angenommen, daß die Liste der Landkreise nach ihren Flächen geordnet ist. Diese Ordnung wird beim Einfügen erhalten.

```
RANGE_OF lk IS Landkreis;
RANGE_OF r IS Regierungsbezirk;
RANGE_OF i IS INDEX; /* Index-Variable */
UPDATE_COMPOBJ r
  INSERT Mitglieder.i lk
  WHERE r.BEZEICHNUNG = 'Lüneburg'
    AND lk.BEZEICHNUNG = 'Celle'
    AND AREA(r.Mitglieder.i-1.GEOMETRIE)
      <= AREA(lk.GEOMETRIE)
    AND AREA(r.Mitglieder.i.GEOMETRIE)
      > AREA(lk.GEOMETRIE);
```

Mit dem "RETRIEVE\_OBJ"-Kommando können Objekte wiedergefunden werden, wobei eine neue temporäre Objektklasse erzeugt wird. Wie üblich werden die Objekte durch Prädikate im "WHERE"-Teil qualifiziert; ihre Attribute mit den zugeordneten Werten werden in der Ziel-Liste angegeben. Im folgenden Beispiel werden Namen von Städten gesucht, in denen SO<sub>2</sub>-Meßpunkte liegen. Dabei soll zu jedem Stadtnamen jeweils nur die höchste Konzentration von SO<sub>2</sub> angegeben werden.

```
RANGE_OF s IS Stadt IN_WINDOW (6,47,14,55); /* BRD */
RANGE_OF m IS SO2_Messung;
RETRIEVE_OBJ INTO erg
  (Stadtname = s.BEZEICHNUNG,
   Konzentration = m.Konzentration)
  WHERE WITHIN(m.GEOMETRIE,s.GEOMETRIE);

RANGE_OF e1 IS erg;
RANGE_OF e2 IS erg;
DELETE_GEOOBJ e1
  WHERE e1.Konzentration < e2.Konzentration
    AND e1.Stadtname = e2.Stadtname;

DISPLAY erg ON TERMINAL;
```

Die Semantik von Anfragen der Datenbanksprache kann als Ausdrücke eines Objektkalküls angegeben werden, der aus dem erweiterten ER-Modell abgeleitet werden kann. Diese Ableitung wird analog zum Tupelkalkül definiert, der deskriptiven relationalen Sprachen zugrunde liegt /Ul82/. Für den ersten Teil der Anfrage aus obigem Beispiel ergibt sich folgender Ausdruck des Objekt-Kalküls:

```
{erg | (∃ s:Stadt) (∃ m:SO2_Messung)
  Stadtname(erg) = BEZEICHNUNG(s) ∧
  Konzentration(erg) = Konzentration(m) ∧
  within(GEOMETRIE(s),{6,47,14,55}) ∧
  within(GEOMETRIE(m),GEOMETRIE(s))}
```

Da die Geodatenbanksprache keine expliziten Quantoren aufweist und implizit lediglich Existenzquantoren auftreten, müssen komplexe Anfragen im mehreren Schritten "berechnet" werden; z.B. wird im obigen Fall die Maximumbildung durch eine "RETRIEVE"- und eine folgende "DELETE"-Anweisung realisiert.

Um eine Karte zu generieren, muß zunächst im fest vorgegebenen Teil des Schemas eine leere Karte eingerichtet werden, indem die benötigten Kartenattribute wie Name, Maßstab usw. als Konstanten eingegeben werden, z.B.:

```
CREATE_MAP('SO_2_Übersicht_in_Städten',1:300000,...);
```

Danach werden geeignet ausgewählte Geoobjekte auf Kartenobjekte abgebildet, die dem Kartenfeld oder dem Kartenrand zugeordnet werden. Dabei stellen die Kartenobjekte im Kartenfeld den eigentlichen Inhalt der Karte dar, während der Rand die Kartenlegende aufnimmt. Beim Übergang von einem Geoobjekt zu einem Kartenobjekt wird zunächst die Geometrie des Geoobjektes an dem "Weltausschnitt"-Fenster der zugeordneten Karte geklippt und dann eine Transformation von den allen Geoobjekten zugrundeliegenden Weltkoordinaten in die Blattkoordinaten der ausgewählten Karte durchgeführt. Neben diesen automatisch ablaufenden Vorgängen kann der Benutzer zahlreiche Attributwerte der neuen Kartenobjekte festlegen, wie Farben, Strichstärken usw.

Im folgenden Beispiel soll eine sehr einfache Übersichtskarte konstruiert werden, in der alle Regierungsbezirke Norddeutschlands als schwarze Flächenumrandungen auftreten sollen; ferner werden Städte als Quadrate dargestellt, deren Farben abhängig von der höchsten SO<sub>2</sub>-Konzentration in der jeweiligen Stadt gewählt werden.

```
/* Regierungsbezirke -> Karte */
RANGE_OF r IS Regierungsbezirk IN_WINDOW (6,52,14,55);
RANGE_OF m IS MAP; /* Norddeutschland */
CREATE_MAPOBJ FROM r
  INSERT_INTO m.FELD
  (FLÄCHEN: FARBE = 'schwarz', FÜLLTYP = 0)
  WHERE m.BEZEICHNUNG = 'SO_2_Übersicht_in_Städten';

/* Städte der 1. Konzentrationsklasse -> Karte (blaue Farbe) */
RANGE_OF s IS Stadt IN_WINDOW (6,52,14,55);
RANGE_OF e IS erg;
CREATE_MAPOBJ FROM s
  INSERT_INTO m.FELD
  (PUNKTE: FARBE = 'blau', MARKERTYP = 4)
  WHERE e.Konzentration <= 0.5
  AND e.Stadtname = s.BEZEICHNUNG
  AND m.BEZEICHNUNG = 'SO_2_Übersicht_in_Städten';
```

Für jede weitere Klassendarstellung, d.h. für jedes weitere Paar bestehend aus Konzentrationsintervall mit zugeordneter Farbe, wird jeweils eine weitere einzelne Anweisung benötigt. Zur Beschriftung könnten in einem abschließenden Kommando Kartenobjekte vom Typ TEXT generiert werden, als deren Inhalte die einzelnen Städtenamen zu wählen wären. Die fertige Karte steht dann in der Datenbank zur weiteren Verarbeitung oder zur Ausgabe zur Verfügung.

Ein größeres Anwendungsbeispiel, das die Konzepte und Möglichkeiten der hier nur kurz vorgestellten "Geo-Datenbanksprache" ausführlich demonstriert, ist in /RN86/ zu finden.

### 3. KLASSEN- UND REGELDEFINITION

Wie im vorherigen Abschnitt dargestellt, werden im Geoobjekt-Modell der Inhalt einer Karte durch Kartenobjekte beschrieben. Diese Kartenobjekte sind eine Abbildung von Geoobjekten in einer Karte, in der Datenbanksprache durch die Operation CREATE\_MAPOBJ realisiert. Der Entwurf einer Karte besteht in der Definition einer großen Anzahl von Abbildungsvorschriften. Durch die im Folgenden vorgeschlagenen Sprachkonstrukte soll der Entwurfsvorgang vereinfacht werden, indem immer wiederkehrende Anweisungsfolgen nur ein einziges Mal parametrisch beschrieben werden müssen. Dabei ziehen wir aus der Tatsache Nutzen, daß Abbildungsvorschriften in der Regel für Gruppen von Karten und nicht nur für Einzelkarten gültig sind. So gelten z.B. für Flurkarten Darstellungsnormen nach DIN 15, DIN 6776 und DIN 18702 (nach /Ha85/).

Abbildungsvorschriften oder -regeln setzen sich aus folgenden Angaben zusammen: "unter welchen Bedingungen wird etwas ausgeführt", d.h. welche Geoobjekte sollen abgebildet werden und für welche Karten soll die Vorschrift verwendet werden, und "welche Aktion wird ausgeführt", d.h. wie sollen die Geoobjekte dargestellt werden.

Die abzubildende Menge der Geoobjekte kann identisch sein mit einer Geoobjektklasse. Durch die einheitliche Darstellung einer Klasse enthält die Karte nur Informationen über die räumliche Orientierung der Objekte. Oft soll jedoch zusätzlich eine Eigenschaft, d.h. ein Attribut des Objektes dargestellt werden. Hierfür wird der Wertebereich des darzustellenden Attributes in Werteklassen aufgeteilt, von der die Darstellung des Objektes in der Karte abhängt. Sowohl quantitative Merkmale, wie z.B. Einwohnerzahlen, als auch qualitative Merkmale, wie Bodenarten, können auf diese Weise aufgeteilt werden. Damit legt man gleichzeitig eine Unterteilung in speziellere Klassen fest, wobei die entstehenden Unterklassen disjunkt sein sollen.

Die Aufteilung der Geoobjektklassen, die Klassifikation, kann durch die Operation CREATE\_CLASS definiert werden. Für die Klassifikation muß ein Name angegeben werden, um sie von anderen Klassifikationen der gleichen Geoobjektklasse unterscheiden zu können. Außerdem werden alle Unterklassen einer Klassifizierung benannt. Sie können über ihren Namen bei der Kartenerstellung angesprochen werden. Zu jeder Unterklasse der Geoobjektklasse wird ein Ausdruck, bestehend aus Einzelwerten oder Intervallen, angegeben, mit dem der Unterklasse ein Wertebereich des zur Klassenbildung benutzten Attributs zugeordnet wird. Ein Beispiel für die Definition der Klassifikation:

```
CREATE_CLASS Waldschadensklassen FOR Waldschäden
  SUBCLASSES VIA geschädigte_Fläche
    keine_Schädigung      : ( .. 0.2)
    geringe_Schädigung    : [0.2 .. 0.35)
    mittlere_Schädigung   : [0.35 .. 0.5)
    starke_Schädigung     : [0.5 .. 0.65)
    absterbender_Wald     : [0.65 .. );
```

Für die Geoobjektklasse "Waldschäden" haben wir eine Klassifikation "Waldschadensklassen" definiert. Die Aufteilung in Unterklassen erfolgt, indem der Wertebereich des Attributes "geschädigte Fläche" in 5 Intervalle aufgeteilt wird. Die Unterklassen sind disjunkt, weil die Intervalle des zur Aufteilung verwendeten Attributes disjunkt sind.

Um angeben zu können, für welche Karte eine Abbildung gelten soll, wird ein Mechanismus zur Gruppierung von Karten eingeführt, indem das Standard-Objekt "Karte" um ein Attribut "Kartenart" erweitert wird. Mit "Kartenart" werden Gruppen von Karten bezeichnet, z.B. Topologische Karte, Geologische Übersichtskarte usw. (vgl. /Wi81/). Durch die Angabe von Wertebereichen für die Kartenattribute "Kartenart" und "Maßstab" wird eine Abbildungsregel für eine Gruppe von Karten definiert.

Die Beschreibung, wie Objekte in Karten dargestellt werden sollen, geschieht, indem die zu erzeugenden Kartenobjekte direkt angegeben werden. Dabei lehnt sich die Spezifikation der Kartenobjekte an die in Abschnitt 2 eingeführte Schreibweise an; für jedes Kartenobjekt werden der Geometriety und die zugehörigen graphischen Attribute definiert.

Für die Operation CREATE\_MRULE benutzen wir als Rahmen eine IF-THEN Konstruktion. Zusätzlich wird ein Regelkopf eingeführt, in dem ein eindeutiger Regelname angegeben und die Geobjektklasse spezifiziert wird, auf die sich die Regel bezieht. Der Regelname identifiziert eine Regel und wird insbesondere beim Löschen einer Regel benutzt.

Im IF- oder Bedingungsteil der Regel werden die Karten und die Geobjekte genauer spezifiziert. Der Anwendungsbereich der Regel für Karten wird durch die Kartenart und den Maßstab definiert. Dabei können mehrere Kartenarten angegeben werden oder auch ein Intervall für den Maßstab. Einzelne Regeln können auch einheitlich für alle Kartenarten gelten, in diesem Fall entfällt die Spezifikation der Kartenart. Analog bedeutet das Auslassen der Maßstabsangabe, daß es keine Einschränkung bzgl. des Maßstabes gibt.

Die Geobjektklasse, für die die Regel benutzt werden soll, ist bereits im Regelkopf spezifiziert. Diese Klasse kann durch die IS\_IN...BY-Anweisung auf Unterklassen beschränkt werden. Es werden dann nur Exemplare der Geobjektklasse, die in den durch IS\_IN angegebenen Unterklassen liegen, in eine Karte abgebildet. Diese Unterklassen müssen vorher durch die mit BY spezifizierte Klassifikation definiert worden sein.

Im THEN-Teil werden die zu erzeugenden Kartenobjekte direkt angegeben. Mehrere Kartenobjekte können für ein Geobjektexemplar zusammen erzeugt werden, wenn z.B. neben der Fläche auch die Bezeichnung in die Karte mit aufgenommen werden soll.

Zur Veranschaulichung dieser Sprachkonzepte führen wir einige Beispiele an. Zunächst greifen wir auf die oben definierte Klassifikation "Waldschadensklassen" zurück:

```
CREATE_MRULE Waldschadenskarte_keine_Schäden (Waldschäden:w)
  IF KARTENART = Waldschadenskarte AND
     MASSTAB >= 1;500000 AND MASSTAB < 1;100000 AND
     w IS_IN keine_Schädigung BY Waldschadensklassen
  THEN MAPOBJECT (FLÄCHEN:
    FARBE = 'hellgrün',
    FULLTYP = 1);
```

Die Schadensklasse "keine\_Schädigung" soll in allen Waldschadenskarten als hellgrüne Fläche dargestellt werden. Für die restlichen Schadensklassen wären analoge Regeln notwendig.



Im nächsten Beispiel gilt eine Regel für eine ganze Geoobjektklasse:

```
CREATE_MRULE
  Städte_in_Waldschadenskarten_mittl_Maßstabs {Stadt:s}
  IF KARTENART = Waldschadenskarte AND
    MASSTAB >= 1:1000000 AND MASSTAB < 1:100000
  THEN MAPOBJECT {FLÄCHEN:
    FÜLLTYP = 2,
    FARBE = 'grau',
    MUSTERTYP = 326},
    MAPOBJECT {TEXT:
    POSITION = CENTRE (s.GEOMETRIE),
    INHALT = s.BEZEICHNUNG);
```

Für die Geoobjekte der Klasse "Stadt" werden zwei Kartenobjekte generiert, eins für die Darstellung der Stadt als grau gemusterte Fläche, und ein Kartenobjekt zur Darstellung des Stadtnamens.

Im dritten Beispiel werden wir eine Regel definieren, die für alle Kartenarten angewendet werden soll:

```
CREATE_MRULE
  Landkreisgrenzen_in_Karten_mittl_Maßstabs {Landkreis:lk}
  IF MASSTAB >= 1:500000 AND MASSTAB < 1:50000
  THEN MAPOBJECT {LINIEN:
    GEOMETRIE = BORDER (lk.GEOMETRIE),
    LINIENTYP = 51,
    BREITENFAKTOR = 0.3,
    FARBE = 'anthrazit'},
    MAPOBJECT {TEXT:
    POSITION = CENTRE (lk.GEOMETRIE),
    INHALT = lk.BEZEICHNUNG);
```

Landkreisgrenzen werden als gestrichelte schwarze Linien dargestellt. In die umschlossene Fläche wird der Name des Landkreises plaziert.

In diesem Abschnitt haben wir nur die Sprachmittel zur Definition von Klassifikationen und Abbildungsregeln genauer erläutert. Dabei wurde auf ebenso notwendige Löscho- und Anfrageoperationen nicht weiter eingegangen. Die Syntax der hier vorgestellten Anweisungen ist im Anhang zusammengestellt.

#### 4. KARTENKONSTRUKTION

Die im letzten Abschnitt vorgestellten Operationen haben rein deklarativen Charakter. Mit der im Folgenden erläuterten Anweisung CONSTRUCT\_MAP soll festgelegt werden, welche Geoobjekte in einem Kartenexemplar enthalten sein sollen. Der Inhalt einer Karte soll jedoch nicht nur deklariert, sondern die Karte auch explizit erzeugt werden, d.h. die durch die Beschreibung von Unterklassen und Abbildungen und durch das Kartenexemplar implizierten Datenbankoperationen müssen generiert und ausgeführt werden. An dieser Stelle werden wir das Sprachelement genauer darstellen, während die Umsetzung in die Geo-Datenbanksprache im nachfolgenden Abschnitt erläutert wird.

In der Anweisung CONSTRUCT\_MAP werden die zu einer Karte gehörenden Datenbankobjekte beschrieben, d. h. ein Exemplar der Klasse KARTE und mehrere Kartenobjekte für die zusammengesetzten Subobjektklassen RAND

und FELD. Den Attributen des Objektes Karte werden explizit Werte zugewiesen. Für den Kartenrand und das Kartenfeld werden die Geoobjektklassen aufgezählt, für die anhand der vorliegenden Abbildungsregeln Kartenobjekte erzeugt werden sollen.

Ein Element dieser Aufzählung kann auf drei verschiedene Arten aufgebaut sein. Wenn nur der Name einer Geoobjektklasse angegeben wird, so bedeutet dies, daß alle Abbildungsregeln für die Kartenobjekterzeugung benutzt werden, die für diese Karte und diese Geoobjektklasse bzw. deren Teilklassen anwendbar sind. Wenn dem Geoobjektnamen mit BY ein Klassifikationsname angefügt wird, werden nur Abbildungsregeln für die Unterklassen dieser Klassifikation verwendet. Durch Angabe einzelner Unterklassen (mit IS\_IN) werden nur die aufgeführten Unterklassen abgebildet. Außerdem kann die Menge der darzustellenden Exemplare - unabhängig von einer Klassifikation - beschränkt werden, indem mit IF eine nachfolgende Bedingung für ein Attribut der Geoobjektklasse angehängt wird.

Im nachfolgenden Beispiel werden die bisher definierten Klassifikationen und Abbildungsregeln verwendet:

```
CONSTRUCT_MAP
  KARTENART = 'Waldschadenskarte',
  BEZEICHNUNG = 'Waldschadenskarte 1984, Lüneburg',
  AUTOR = 'H.Müller',
  MASSTAB = 1:400000,
  ...
  WELTAUSSCHNITT = (6, 52, 14, 55),
  FELD =
    (Landkreis,
     Stadt IF Bevölkerung > 5000,
     Waldschäden BY Waldschadensklassen IF Erhebungsjahr = 1984);
```

Mit dieser Anweisung wird eine Karte zum Thema Waldschaden erstellt, aus der über alle Baumarten zusammengefaßt der Anteil der geschädigten Fläche an der Gesamt-Waldfläche ersichtlich ist. Die Karte umfaßt die Darstellung der Landkreise und der größeren Städte. Waldgebiete werden (wie in /FN85/) entsprechend der Einordnung in eine Waldschadensklasse farblich dargestellt.

## 5. UMSETZUNG DER KARTENKONSTRUKTION

Dieser Abschnitt skizziert die automatische Übersetzung von Kartenkonstruktionsanweisungen in die Datenmanipulationssprache, die in Abschnitt 2 vorgestellt wurde. Die durch das geowissenschaftliche Datenbanksystem bislang realisierte Benutzerschnittstelle wird damit um diese recht mächtige Anweisung erweitert.

Dem Übersetzer müssen die Klassendefinitionen und Regeln, wie sie in den letzten Abschnitten eingeführt wurden, in geeigneten Datenstrukturen zur Verfügung stehen, da eine vorgegebene CONSTRUCT\_MAP-Anweisung durch Anwendung der jeweils passenden Regeln in mehrere CREATE\_MAPOBJ-Kommandos umgesetzt wird, wobei die verschiedenen Regel- und Klassenprädikate in die WHERE-Liste der Kartenobjekt-Generierungsanweisungen einfließen müssen. Solche CREATE\_MAPOBJ-Anweisungen sind zu erzeugen für:

- alle Geoobjekt-Angaben einer CONSTRUCT\_MAP-Anweisung
  - innerhalb einer Geoobjekt-Angabe für alle Regeln, die für die aktuelle Geoobjekt-Angabe passen
    - innerhalb einer Regel für alle Kartenobjekt-Klassen, die in dieser Regel aufgeführt werden.

Anhand eines Beispiel soll diese Erzeugung von Kartenobjekten erläutert werden. Es sei folgende "Waldschadenskarte" zu generieren:

```

CONSTRUCT_MAP
  KARTENART = 'Waldschadenskarte',
  BEZEICHNUNG = 'Nordd. Waldschäden',
  ....
  WELTAUSSCHNITT = (6,52,14,55)
  FELD =
    (Stadt IF Bevölkerung > 10000,
     Waldschäden
      IS_IN (starke_Schädigung, mittlere_Schädigung)
      BY Waldschadensklassen );

```

Aufgrund der Regel "Städte in Waldschadenskarten mittleren Maßstabs" (vgl. Abschnitt 3) sind Kartenobjekte der Klassen FLÄCHEN und TEXT aus Geoobjekten der Klasse "Stadt" zu generieren. Für die Geoobjekt-Klasse "Waldschäden" hingegen sind zwei Regeln anwendbar, die jeweils die Erzeugung von Kartenobjekten der Klasse FLÄCHEN vorsehen. Insgesamt liefert ein entwickelter Algorithmus, der aus Platzgründen hier nicht angegeben werden kann, bei der Eingabe der o.a. CONSTRUCT\_MAP-Anweisung sowie den im vorigen Abschnitt aufgeführten konkreten Regeln und Klasseneinteilungen folgende Übersetzung:

```

/* 1 */ CREATE_MAP('Nordd. Waldschäden',...,{6,52,14,55});

RANGE_OF m IS MAP;

/* 2 */ RANGE_OF g1 IS Stadt IN_WINDOW (6,52,14,55);

/* 3 */ CREATE_MAPOBJ FROM g1
INSERT INTO m.FELD
  (FLÄCHEN: FÜLLTYP = 2,
   ....
   MUSTERTYP = 326)
WHERE m.BEZEICHNUNG = 'Nordd. Waldschäden'
AND g1.Bevölkerung > 10000;

/* 4 */ CREATE_MAPOBJ FROM g1
INSERT INTO m.FELD
  (TEXT: POSITION = CENTRE(g1.GEOMETRIE),
   INHALT = g1.BEZEICHNUNG)
WHERE m.BEZEICHNUNG = 'Nordd. Waldschäden'
AND g1.Bevölkerung > 10000;

RANGE_OF g2 IS Waldschäden IN_WINDOW (6,52,14,55);

/* 5 */ CREATE_MAPOBJ FROM g2
INSERT INTO m.FELD
  (FLÄCHEN: FARBE = 'orange',
   FÜLLTYP = 1)
WHERE m.BEZEICHNUNG = 'Nordd. Waldschäden'
AND (g2.geschädigte_Fläche >= 0.35
AND g2.geschädigte_Fläche < 0.5);

```

```

/* 6 */ CREATE_MAPOBJ FROM g2
INSERT INTO m.FELD
  (FLÄCHEN: FARBE = 'hellrot',
   FÜLLTYP = 1)
WHERE m.BEZEICHNUNG = 'Nordd. Waldschäden'
  AND {g2.geschädigte_Fläche >= 0.5
  AND g2.geschädigte_Fläche < 0.65};

```

Es wird also zunächst eine Karte mit den entsprechenden Konstanten angelegt (1). Die anschließenden Deklarationen der Geoobjekt-Variablen (2) berücksichtigen die Tatsache, daß nur Geoobjekte von Interesse sind, deren Geometrien innerhalb des Weltausschnitts der Karte liegen.

Bei den WHERE-Listen der CREATE\_MAPOBJ-Anweisungen (3-5) sind drei verschiedene Prädikat-Klassen zu erkennen, die, durch logisches Und verknüpft, jeweils den Gesamtqualifikationsteil ausmachen. Diese drei Prädikat-Arten, von denen nur die erste der unten angeführten auftreten muß, lassen sich in ihrer Funktion wie folgt charakterisieren: es sind Prädikate, die

- der Qualifikation der Karte dienen ("m.BEZEICHNUNG =...")
- eine optionale zusätzlichen Qualifikation von Geoobjekten wiedergeben, z.B. "gl.Einwohner > 10000"
- durch die optionale Angabe von Unterklassen entstehen, z.B. "g2.geschädigte\_Fläche >= 0.5 AND g2.geschädigte\_Fläche < 0.65".

Die so erzeugten Kartenobjekt-Generierungskommandos werden zusammen mit der CREATE\_MAP-Anweisung und den Objektvariablen-Deklarationen dem geowissenschaftlichen Datenbanksystem übergeben und von diesem ausgeführt. Danach steht die neue Karte innerhalb der Datenbank zur Ausgabe oder weiteren Verarbeitung zur Verfügung. Hier sind z.B. gezielte Veränderungen von einzelnen Kartenobjekten denkbar, die Ausnahmen von Regeln darstellen, etwa eine eigene Farbe für eine auszuzeichnende Stadt. Solche Änderungen an einer bereits konstruierten Karte müssen direkt durch einzelne Anweisungen der Datenbanksprache per Dialog realisiert werden.

## 6. STAND DER REALISIERUNG UND AUSBLICK

Aufbauend auf einem geowissenschaftlichen Datenbanksystem haben wir in den letzten Abschnitten Sprachelemente vorgeschlagen, mit deren Hilfe Kartenelemente regelgesteuert erzeugt werden können. Die Struktur der zulässigen Regeln wurde allerdings sehr eng gehalten, weshalb die vorliegende Arbeit lediglich als kleiner Schritt in Richtung auf eine automatische Erzeugung realistischer und somit ungleich komplexerer Karten, als die von uns vorausgesetzten, zu sehen ist.

Ein erster partieller Prototyp des in Abschnitt 2 vorgestellten Geodatenbanksystems ist implementiert worden (/Er86, St86, Wa86/) und wird zur Zeit getestet. Die Basis dieser Implementierung bildet eine relationale Datenbankmaschine IDM 500 mit der QUEL-ähnlichen Sprache IDL. Durch die erzwungene Verteilung der einzelnen Objektexemplare auf zahlreiche flache Implementierungsrelationen müssen beim Zugriff auf Objekte jeweils, für den Benutzer unsichtbar, viele implizite relationale Verbunde generiert werden. Es ist deshalb abzusehen, daß die Antwortzeiten komplexer Operationen, wie der Kartengenerierung, recht problematisch sein werden. Abhilfe erwarten wir hier durch die Im-

plementierung des zweiten Prototyps auf der Basis eines "Geo-Kerns" /DOPS85, SW86/, der Objekte auf NF<sup>2</sup>-Relationen /SS86/ abbildet und räumliche Zugriffspfade für geometrische Datentypen bietet. Dies ist für die bei der Kartenerstellung sehr häufig auftretende Suche von Objekten innerhalb von "Fenstern" (Weltausschnitts-Bildung) besonders wichtig.

Parallel zum zweiten Prototypen werden wir eine erste Version der beschriebenen Kartenentwurfskomponente erstellen. Als Vorbereitung hierfür müssen geeignete Speicherungsverfahren für die Beschreibung der Klassifikation und der Abbildungsregeln entworfen werden, die für effizientes Auffinden der Beschreibungen bei der Kartenerstellung ausgelegt sind. Ein weiterer Gesichtspunkt für die Regelverwaltung wird die geeignete Unterstützung einer Konsistenzüberprüfung der Regeln sein. Dabei verstehen wir Regelkonsistenz so, daß es nicht mehrere Regeln geben darf, durch die eine Eigenschaft eines Geobjektes in einer Karte widersprüchlich dargestellt wird.

Karten, die mit unserer Komponente erstellt werden, haben den Status von Entwürfen. Um fertige Karten zu erhalten, bedarf es der Kartenverbesserung (nach /Br83/), bestehend aus der Platzierung der Beschriftung und dem Auflösen von Überlappungen auf der Karte. Zur Zeit wird dieser Arbeitsschritt weitgehend manuell durchgeführt, es gibt jedoch Ansätze zur Automation. So werden z.B. in /FA84, PBHH85/ Expertensysteme zur automatischen Beschriftung und in /RJ85/ ein Expertensystem zur Kartenerstellung vorgestellt. In /RFB86/ wird ein Überblick und eine Einschätzung von Expertensystemen in den Geowissenschaften gegeben. Es ist zu hoffen, daß unser Ansatz des Kartenentwurfs so erweitert werden kann, daß auch der Prozeß der Kartenverbesserung weitgehend automatisiert durchführbar wird.

Die Autoren möchten an dieser Stelle ihrem Kollegen U.W. Lipeck für seine Anregungen und Kommentare zu dieser Arbeit danken.

## LITERATUR

- /Br83/ Brassel, K.: Geographisch-kartographische Informationsverarbeitung. Output (1983), Heft 7, 21-28
- /Ch76/ Chen, P.P.: The Entity-Relationship Model - Toward a Unified View of Data. ACM ToDS 1 (1976), 9-36
- /DOPS85/ Deppisch, U./ Obermeit, V./ Paul, H.-B./ Schek, H.-J./ Scholl, M./ Weikum, G.: The Storage Subsystem of a Database Kernel System. Techn. Report DVSI-1985-T1, TH Darmstadt.
- /Er86/ Ernesti, B.: Realisierung eines Geo-Datenbank-Prototyps; Der Geometrieprozessor mit einfacher graphischer Ausgabe. Diplom-Arbeit, TU Braunschweig 1986
- /FA84/ Freeman, H./ Ahn, J.: AUTONAP - An Expert System for Automatic Map Name Placement. in: Proc. 1st Int. Symposium on Spatial Data Handling (1984), 544-569
- /FN85/ Fischer, D./ Nouhuys, J.V.: Stand und Entwicklung der graphischen Datenverarbeitung in der Umweltberichterstattung. CAD-Kartographie, Schilcher, M. (Hrsg.), 1985, 205-221
- /Ha85/ Hake, G.: Kartographie II. 3. Aufl., de Gruyter, 1985

- /LN86/ Lipeck,U.W./ Neumann,K.: Modelling and Manipulating Objects in Geoscientific Databases. Proc. 5th Int. Conf. on the Entity-Relationship Approach. S.Spaccapietra (ed.), Dijon 1986, 105-124
- /PBHH85/ Pfefferkorn,C./ Burr,D./ Harrison,D./ Heckman,B.: ACES: A Cartographic Expert System. 7th Int. Symposium on Computer Assisted Cartography, Washington, D.C., 1985, 399-407.
- /RJ85/ Robinson,G./ Jackson,M.: Expert Systems in Map Design. 7th Int. Symposium on Computer Assisted Cartography, Washington, D.C., 1985, 430-439.
- /RFB86/ Robinson,V.B./ Frank,A.U./ Blaze,M.: An Assessment of Expert Systems Applied to Problems in Geographic Information Systems. ASCE Specialty Conference on Integrated Geographic Information Systems, 1986.
- /RN86/ Ramm,I./ Neumann,K.: Anwendungen auf Geo-Datenbanken: Ein Beispiel. Interner Bericht, TU Braunschweig 1986
- /RNLE85/ Ramm,I./ Neumann,K./ Lipeck,U.W./ Ehrich,H.-D.: Eine Benutzerschnittstelle für geowissenschaftliche Datenbanken. Informatik-Bericht 85-06, TU Braunschweig 1985
- /Ro85/ Rohdenburg,H. (Hrsg.): Antrag auf Einrichtung des Sonderforschungsbereiches "Wasser- und Stoffdynamik von Agrar-Ökosystemen". TU Braunschweig 1985
- /SS86/ Schek,H.-J./ Scholl,M.H.: The Relational Model with Relation-Valued Attributes. Information Systems 11 (1986), 137-147
- /St86/ Stahs,T.: Realisierung eines Geo-Datenbank-Prototyps: Der Übersetzer. Diplom-Arbeit, TU Braunschweig 1986
- /SW86/ Schek,H.-J./ Waterfeld,W.: A Database Kernel System for Geoscientific Applications. in: Proc. 2nd Int. Symposium on Spatial Data Handling (1986)
- /SWKH76/ Stonebraker,M./ Wong,E./ Kreps,P./ Held,G.: The Design and Implementation of INGRES. ACM ToDS 1 (1976), 198-222
- /Ul82/ Ullman,J.D.: Principles of Database Systems. 2nd ed., Computer Science Press, Rockville (Md.) 1982
- /Vi85/ Vinken,R.: Digitale Geowissenschaftliche Kartenwerke - ein neues Schwerpunktprogramm der Deutschen Forschungsgemeinschaft. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 95 (1985), 163-173
- /Wa86/ Warnebold,P.: Realisierung eines Geo-Datenbank-Prototyps: Der Benutzerdialog mit syntaktischer und semantischer Analyse. Diplom-Arbeit, TU Braunschweig 1986
- /Wi81/ Wilhelmy,H.: Kartographie in Stichworten. 4.Aufl., F. Hirt, 1981

## ANHANG: Syntax der Klassen- und Regeldefinition sowie der Kartengenerierung

Im folgenden wird die Syntax der Definitionen für Klassen und Regeln sowie der Anweisung zur Kartengenerierung vorgestellt. Wir bedienen uns dabei einer erweiterten BNF-Grammatik; ein Term in eckigen Klammern [term] ist als optional zu betrachten, Terme in geschweiften Klammern {term} können beliebig oft - auch gar nicht - wiederholt werden, die spitzen Klammern bedeuten, daß genau ein Term ausgewählt werden muß <term 1 | term 2 [...] term n>. Nichtterminalsymbole sind kleingeschriebene Zeichenketten, die Verbindungsstriche enthalten können, während Terminalsymbole meist durch grosse Buchstaben gebildet werden. Die Grammatik soll lediglich einen Überblick vermitteln, es sind deshalb nicht alle Nichtterminalsymbole bis auf Terminalsymbole ableitbar, das gilt insbesondere für alle "-namen" und "-konstanten". Einige Symbole, wie etwa "ausdruck" sind in der Grammatik aus /RNLE85/ erklärt.

```
klassen-definition ::=
    CREATE_CLASS klassen-name FOR geoobjekt-klassen-name
    SUBCLASSES VIA geoobjekt-attribut-name
        unterklassen-name : wertebereich {, wertebereich}
        {unterklassen-name : wertebereich {, wertebereich}}
        [unterklassen-name : OTHERWISE] ;

wertebereich ::= konstante | intervall

intervall ::= < ( . | ( ) < konstante.. |
                konstante..konstante |
                ..konstante >
            < .) | ) >

regel-definition ::=
    CREATE_RULE regel-name ( geoobjekt-klassen-name :
                            variablen-name )
        [IF regel-bedingung THEN]
        kartenobjekt-definition
        {, kartenobjekt-definition} ;

regel-bedingung ::=
    <kartenart-prädikat [AND maßstabs-prädikat]
        [AND unterklassen-prädikat] |
    maßstabsprädikat [AND unterklassen-prädikat] |
    unterklassen-prädikat>

kartenartprädikat ::=
    KARTENART = { kartenart-konstante
                {, kartenart-konstante} )

maßstabs-prädikat ::=
    MASSTAB vgl-op maßstab-konstante
    {<AND | OR> MASSTAB vgl-op maßstab-konstante}
```

```

unterklassen-prädikat ::=
    variablen-name IS_IN ( unterklassen-name
                          {, unterklassen-name} )
    BY klassen-name

kartenobjekt-definition ::=
    MAPOBJECT { <PUNKTE | LINIEN | FLACHEN | TEXT> :
    kartenobjekt-attribut-name = ausdruck
    {, kartenobjekt-attribut-name = ausdruck} }

karten-generierung ::=
    CONSTRUCT_MAP
    KARTENART = kartenart-konstante ,
    BEZEICHNUNG = bezeichnung-konstante ,
    AUTOR = autor-konstante ,
    MASSTAB = maßstab-konstante ,
    WELTAUSSCHNITT = weltausschnitt-konstante ,
    BLATT = blatt-konstante ,
    BLATTAUSSCHNITT = blattausschnitt-konstante
    FELD = { geobjekt-angabe
            {, geobjekt-angabe} }
    [RAND = { geobjekt-angabe
            {, geobjekt-angabe} } ] ;

geobjekt-angabe ::=
    geobjekt-klassen-name
    [[IS_IN ( unterklassen-name
            {, unterklassen-name} )]
    BY klassen-name]
    [IF geobjekt-attribut-name vgl-op konstante]

```