



## Objektkatalog für das Straßen- und Verkehrswesen

### Modellierung der allgemeinen Geometrieobjekte

Version: 0.06  
Datum: 26.09.2000  
Status: vorgelegt  
Dateiname: N0011.doc  
Pfad: /...  
Verantwortlich: Dietmar König

#### **OKSTRA-Pflegestelle**

interactive instruments GmbH  
Trierer Straße 70-72  
53115 Bonn

Herr Clemens Portele  
Tel. 0228 91410 73  
Fax 0228 91410 90  
Email [portele@interactive-instruments.de](mailto:portele@interactive-instruments.de)

#### **Im Auftrag von**

Bundesanstalt für Straßenwesen  
ZD - OKSTRA  
Brüderstraße 53  
51427 Bergisch Gladbach

Herr Alfred Stein  
Tel. 02204 43 354  
Fax 02204 43 673  
Email [stein@bast.de](mailto:stein@bast.de)



# 0 Allgemeines

## 0.1 Inhaltsverzeichnis

<b>0 Allgemeines</b> .....	<b>2</b>
0.1 Inhaltsverzeichnis .....	2
0.2 Abkürzungen und Definitionen .....	2
0.3 Abbildungsverzeichnis .....	3
0.4 Tabellenverzeichnis .....	3
0.5 Bezüge .....	3
0.6 Änderungen .....	3
0.7 Bearbeitungsvermerke .....	4
<b>1 Zweck des Dokuments</b> .....	<b>5</b>
1.1 Leserkreis .....	5
1.2 Kernaussagen des Inhalts .....	5
<b>2 NIAM-Modellierung</b> .....	<b>6</b>
2.1 Allgemeine Geometrieobjekte .....	6
2.1.1 Erläuterungen .....	7
2.2 Digitales Geländemodell .....	9
2.2.1 Erläuterungen .....	9
2.3 Beschriftung .....	10
2.3.1 Erläuterungen .....	10
<b>3 EXPRESS-Schema</b> .....	<b>12</b>
3.1 Auszüge anderer Schemata .....	15
<b>4 SQL-Schema</b> .....	<b>19</b>
<b>5 Anhang: Geometrie-Schema</b> .....	<b>26</b>
5.1 Erläuterungen .....	27

## 0.2 Abkürzungen und Definitionen

OKSTRA	Objektkatalog für das Straßen- und Verkehrswesen
DGM	Digitales Geländemodell
FGSV	Forschungsgesellschaft für Straßen- und Verkehrswesen e.V.
DA001	Datenart 001 (Datenart der FGSV)
AdV	Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland
RAS-Verm	Richtlinien für die Anlage von Straßen, Teil: Vermessung
ALK/ALKIS	Automatisierte Liegenschaftskarte (ALK) und ihre Weiterentwicklung ALKIS



### 0.3 Abbildungsverzeichnis

Abbildung 1 - Allgemeine Geometrieobjekte.....	6
Abbildung 2 - Digitales Geländemodell .....	9
Abbildung 3 - Beschriftung.....	10
Abbildung 4 - Geometrie-Schema des OKSTRA.....	26

### 0.4 Tabellenverzeichnis

(keine Tabellen enthalten)

### 0.5 Bezüge

Dokument	Bemerkungen
OKSTRA-Website	<a href="http://www.okstra.de">http://www.okstra.de</a>
T0001	Einführung in die NIAM-Technik, zu finden auf der OKSTRA-Website unter <i>Datenschema</i>
D008-1000	Teilschema 008: Entwurf / NIAM, zu finden auf der OKSTRA-Website unter <i>Datenschema</i>
D018-1000	Teilschema 018: Geometrie / NIAM, zu finden auf der OKSTRA-Website unter <i>Datenschema</i>
geometr.cfl	NIAM-Diagramm zum Geometrie-Schema des OKSTRA, zu finden im Dokument D018-1000
N0012	Vorschlag zur Übertragung eines Entwurfes, zu finden auf der OKSTRA-Website unter <i>Dokumente</i>

### 0.6 Änderungen

Name	Datum	Kapitel	Bemerkungen	Bearbeiter
N0007	29.02.2000	alle	Dokument erstellt	Dietmar König
N0011	06.07.2000	alle	Dokument aufgrund der eingegangenen Anmerkungen fortgeschrieben	Clemens Portele Dietmar König
N0011	28.07.2000	alle	Dokument aufgrund der Ergebnisse der Sitzung am 18.Juli in Frankfurt überarbeitet	Clemens Portele Dietmar König
N0011	03.08.2000	alle	Geometriedarstellung überarbeitet	Dietmar König
N0011	22.08.2000	alle	Dokument nach Rücksprache mit Herrn Leverenz ergänzt und überarbeitet	Dietmar König
N0011	21.09.2000	alle	Überarbeitung, Vorlage auf nächster Sitzung der BG10	Dietmar König
N0011	26.09.2000	3, 4	geringfügige Änderungen bei Bezeichnern	Dietmar König

	<b>Objektkatalog für das Straßen- und Verkehrswesen</b> <b>Modellierung der allgemeinen Geometrieobjekte</b>	<b>Seite: 4 von 27</b> <b>Name: N0011</b> <b>Stand: 26.09.2000</b>
--	---	--

## **0.7 Bearbeitungsvermerke**

keine



# 1 Zweck des Dokuments

## 1.1 Leserkreis

Dieses Dokument richtet sich an alle Experten des Bereichs "Entwurf" im Straßen- und Verkehrswesen sowie interessierte Mitglieder der BG10. Es kann jedoch auch für Experten anderer Fachbereiche von Interesse sein, die geometrische Informationen mit dem OKSTRA austauschen wollen.

Vorausgesetzt werden Kenntnisse

- des OKSTRA-Standards in seiner aktuellen Fassung,
- zum OKSTRA und seinen Regularien (siehe auch die OKSTRA-Website <http://www.okstra.de>).

Die Diagramme in diesem Dokument sind in der NIAM-Technik erstellt. Eine Einführung in diese Technik ist in dem Dokument T0001 auf der OKSTRA-Website (unter *Datenschema*) zu finden. Das im NIAM-Diagramm in Kapitel 5 referenzierte Diagramm zum Geometrie-Schema des OKSTRA aus dem OKSTRA-Forschungsprojekt ist im Dokument D018-1000 zu finden (ebenfalls unter *Datenschema*, Teilschema 018: Geometrie, NIAM) bzw. hier auch (in Auszügen) als Anhang im Kapitel 5. Siehe hierzu auch 0.5 Bezüge.

## 1.2 Kernaussagen des Inhalts

Das Dokument stellt einen Vorschlag zur Erweiterung des OKSTRA dar. Modelliert werden vor allem das allgemeine Punktobjekt, das allgemeine Linienobjekt und das allgemeine Flächenobjekt. Diese Objekte erweitern die Möglichkeiten des OKSTRA zum Austausch von geometrischen Informationen.

Der OKSTRA stellt eine einheitliche und vernetzte fachliche Modellierung verschiedener Bereiche des Straßen- und Verkehrswesens zur Verfügung. Der Fokus liegt dabei auf der fachlichen Bedeutung der Objekte. Diese Fachobjekte können dann Geometrie als Eigenschaft tragen.

Für den Austausch von rein oder vorwiegend geometrischen Daten, z.B. komplexen Zeichnungen aus dem Entwurf, muss dieser Ansatz erweitert werden. Falls die fachliche Bedeutung der Geometrie nicht oder noch nicht festgelegt werden kann, so kann sie mit Hilfe der allgemeinen Punkt-, Linien- oder Flächenobjekte als eigenständige Geometrie dargestellt und mit einer Erläuterung versehen über den OKSTRA ausgetauscht werden. Diese Objekte bilden sozusagen Container für Informationen, die ansonsten innerhalb des OKSTRA nicht eigenständig existieren können.

Gedacht ist bei diesen Erweiterungen vor allen Dingen an geometrische Elemente aus dem Bereich des Entwurfs.



## 2 NIAM-Modellierung

### 2.1 Allgemeine Geometrieobjekte

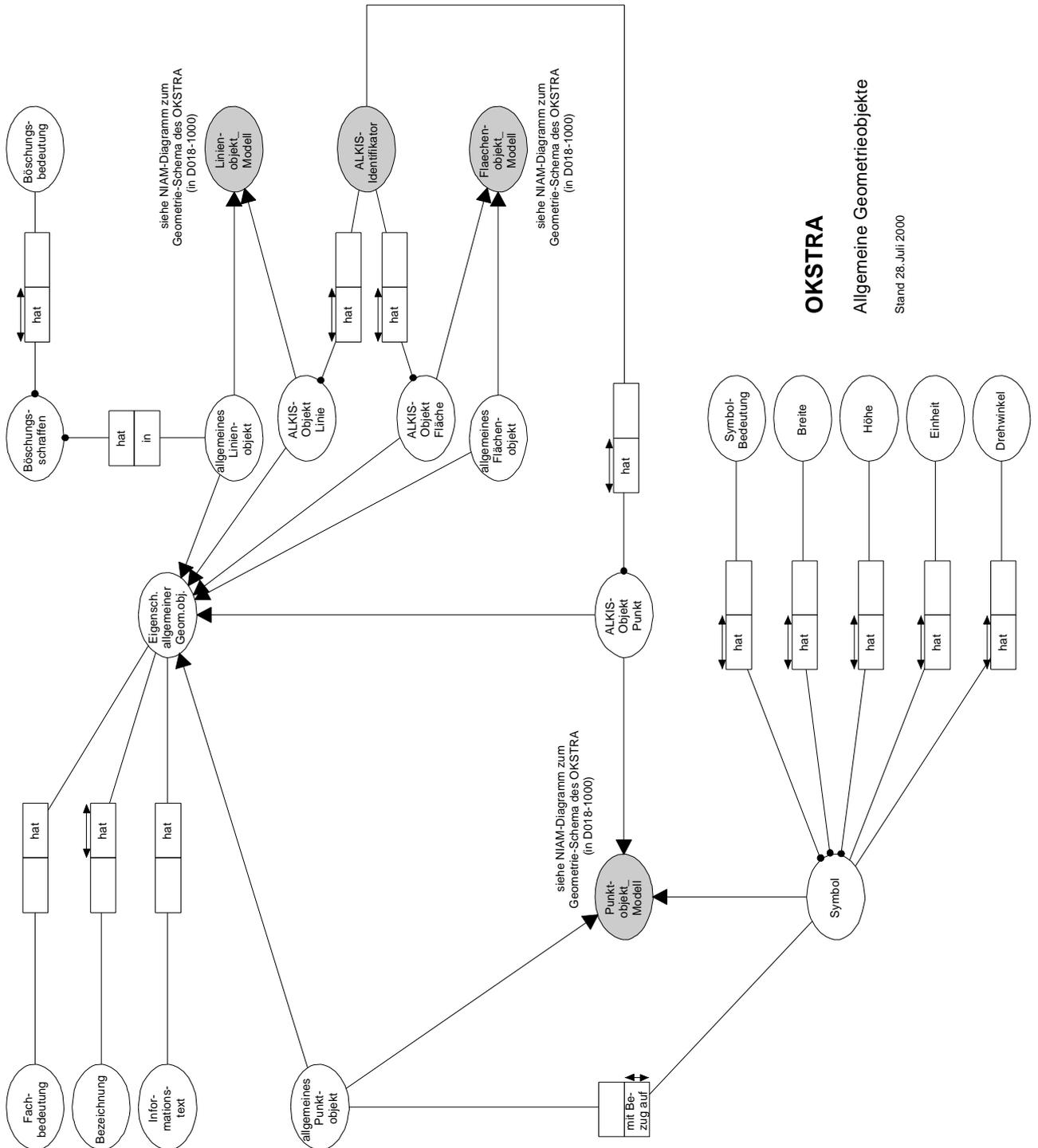


Abbildung 1 - Allgemeine Geometrieobjekte



### 2.1.1 Erläuterungen

Die grau hinterlegten Entities in obigem NIAM-Diagramm (*Punktobjekt\_Modell*, *Linienobjekt\_Modell* und *Flächenobjekt\_Modell*) stammen aus dem Geometrie-Schema des OKSTRA (siehe auch Kapitel 5 Anhang: Geometrie-Schema) und realisieren punktförmige, linienförmige bzw. flächenförmige Eigenschaften fachlicher Entities im OKSTRA. Sie dienen zur einheitlichen Anbindung von geometrischen (und topologischen) Eigenschaften an diese Entities.

Die hier modellierten *allgemeinen Punkt*-, *Linien*- bzw. *Flächenobjekte* erben aus diesen Supertypen und sind damit Spezialisierungen von diesen. Das bedeutet sie tragen zusätzlich zu ihren eigenen Eigenschaften auch die Attribute und Relationen, die der entsprechende Supertyp trägt.

Ein *Punktobjekt\_Modell* hat eine (im Geometrie-Schema dargestellte) Relation zu einem *Punkt*, der wiederum mit seinen Koordinaten und Informationen zum verwendeten Koordinaten-System im Objekt *direct\_position* beschrieben ist. Handelt es sich bei dem Punkt um einen gerechneten Punkt (im Gegensatz zu einem vermessenen Punkt), so kann der Subtyp *gerechneter Punkt* verwendet werden. Das modellierte allgemeine Punktobjekt erhält so auf dem im OKSTRA üblichen Weg Zugang zu einer punktförmigen Geometrie. Für das allgemeine Linienobjekt und das allgemeine Flächenobjekt gelten diese Erläuterungen analog.

Die drei Klassen *allgemeines Punkt*-, *Linien*- und *Flächenobjekt* erben von einer Klasse *Eigenschaften allgemeiner Geometrie-Objekte*, die folgende Attribute besitzt (alle optional):

- Fachbedeutung (Textattribut, multipel)
- Bezeichnung (Textattribut, nicht multipel, eindeutig für alle allgemeinen Geometrie-Objekte mit derselben Fachbedeutung),
- Informationstext (Textattribut, multipel).

Zum Gebrauch dieser neu definierten Entities im OKSTRA ist folgende Grundregel zu beachten:

Ist die fachliche Bedeutung einer Geometrie bekannt und existiert ein zugehöriges fachliches Entity im OKSTRA, so ist dieses zur Darstellung zu verwenden. Nur in Fällen wo keine solche Zuordnung möglich ist sollen die allgemeinen *Punkt*-, *Linien*- oder *Flächenobjekte* verwendet werden.

Diese Ergänzung des OKSTRA um eigenständige Geometrieobjekte - und nichts anderes stellen die neu definierten Entities dar - sollte nur als Zwischenlösung angesehen werden, bis geeignete fachliche Modellierungen durchgeführt und in den OKSTRA aufgenommen worden sind.

Die genaue Festlegung der Anbindung an ALKIS erfolgt später. Hier werden zunächst Objekte *ALKIS-Objekt Punkt*, *ALKIS-Objekt Linie* und *ALKIS-Objekt Fläche* definiert, die einen ALKIS-Objekt-Identifikator als Attribut tragen. Der Bereich Vermessung wird parallel fachlich modelliert. Die allgemeinen Geometrieobjekte dienen bis dahin als Übergangslösung für den Datenaustausch. Sinnvollerweise beschränkt man aber eine Modellierung von ALKIS-Objekten auf Verbindungs-Objekte, die es über einen Verweis auf den ALKIS-Objektidentifikator (Fachdatenverbindung) ermöglichen, deren Eigenschaften zu erfragen. Auf diese Weise wird eine Duplizierung des ALKIS-Modells im OKSTRA vermieden. Für den Austausch von und Zugriff auf ALKIS-Daten sollten grundsätzlich die Standards der AdV Anwendung finden.

Diese ALKIS-(Verbindungs-)Objekte werden nicht von den allgemeinen Geometrieobjekten abgeleitet, sondern eigenständig modelliert da ihre fachliche Bedeutung durch den Bezug zu einem



(externen) ALKIS-Objekt bekannt ist. Dies ist bei den allgemeinen Geometrieobjekten i.a. gerade nicht der Fall.

Die neue Objektklasse *Symbol* erbt von *Punktobjekt\_Modell* und besitzt eine Relation (Kardinalität 0..1) zum allgemeinen Punktobjekt. Diese ist gesetzt, wenn das ausgestaltende Symbol einem fachlich bestimmten allgemeinen Geometrie-Objekt entspricht und z.B. bei einer Löschung oder Verschiebung mitgezogen werden soll. Symbol-Objekte zu einem allgemeinen Punktobjekt müssen nur erzeugt werden, wenn das Symbol

- an einer anderen Position liegen soll oder
- nicht den Vorgaben der RAS-Verm entspricht.

Anders gesagt: Als Default für die zeichnerische Ausgestaltung gilt die RAS-Verm. Soll das entsprechende Symbol von der Lage her verschoben werden oder ein anderes Symbol gewählt werden, so ist dem allgemeinen Geometrieobjekt ein Objekt der Klasse *Symbol* zuzuordnen.

Jedes Symbol trägt die folgenden Attribute:

- Eine Symbolbedeutung (Text, eindeutiges Pflichtattribut), die die zeichnerische Ausgestaltung festlegt. Als Bedeutungen werden vorerst die RAS-Verm-Namen vordefiniert.
- Breite und Höhe (Float, eindeutige Pflichtattribute). Die Angaben entsprechen der Bounding Box vor Anwendung des Drehwinkels.
- Einheit (Text) bezeichnet die Einheit von Breite und Höhe. Der Wert entspricht entweder "Welt" oder "mm". Fehlt die Angabe, so wird "Welt" angenommen. Breite und Höhe sind bei "Welt" in derselben Einheit wie die zugehörige *direct\_position* (siehe Geometrie-Schema des OKSTRA) angegeben. Bei "mm" handelt es sich um Kartenmillimeter.
- Drehwinkel des Symbols von  $-\pi$  bis  $+\pi$ . 0 bedeutet keine Drehung. Siehe hierzu die Festlegungen im OKSTRA-Teilschema zum Entwurf (Dokument D0008-1000).

Dies deckt nicht alle Anforderungen der Ausgestaltung ab. Die zeichnerische Ausgestaltung ist jedoch nicht Thema des OKSTRA. Sie liegt in der Zuständigkeit der Anwendungsprogramme. (siehe zu dieser Fragestellung auch das Dokument N0012)

*Böschungsschraffen* werden als komplexe Variante eines allgemeinen Geometrie-Objekts als eigene Objektklasse modelliert. Sie dienen zur Übertragung der Schraffen und Kehlen. Böschungsschraffen

- tragen eine Relation zu mindestens einem allgemeinen Linienobjekt und
- haben genau eine Böschungsbedeutung (Text, Pflichtattribut, nicht multipel), die die inhaltliche Bedeutung festlegt.
- Die zugehörigen Linien bilden die Schraffen einer Böschung.

Eine Aktualisierung der Böschung als ganzes wird dadurch nicht unterstützt. Einzelne Schraffen oder Kehlen können bearbeitet werden.

Dieser Datenumfang deckt die Anforderungen an die allgemeinen Geometrieobjekte im Rahmen der Übertragung eines Entwurfs ab.





## 2.3 Beschriftung

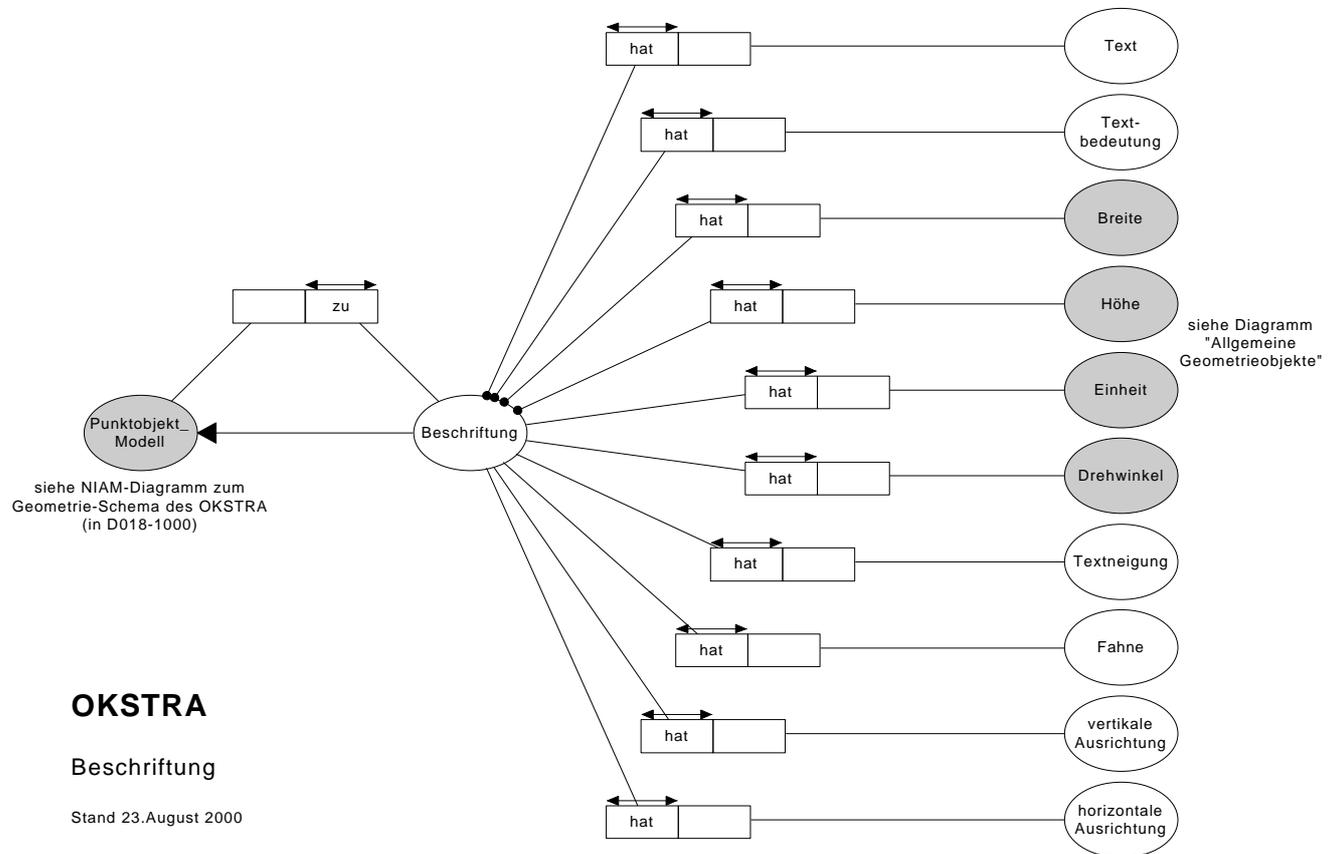


Abbildung 3 - Beschriftung

### 2.3.1 Erläuterungen

Die neue Objektklasse *Beschriftung* erbt von *Punktobjekt\_Modell*. Sie besitzt eine Relation (Kardinalität 0..1) zum *Punktobjekt\_Modell*. Diese ist gesetzt, wenn die Punktposition der *Beschriftung* nicht mit dem Bezugspunkt der *Beschriftung* übereinstimmt und eine Fahne gezeichnet werden soll.

Jede *Beschriftung* trägt die folgenden Attribute:

- Den Text selbst (Text, Pflichtattribut, nicht multipel). Der Text wird RTF-codiert, so dass auch die Textausgestaltung mit abgelegt werden kann. RTF wird im Rahmen von OKSTRA(-CTE) nur mit folgenden Funktionalitäten verwendet:
  - Textinhalt mit Umlauten und Sonderzeichen (Promille, Unendlich, Delta etc.)
  - Schriftart
  - Schriftgröße
  - Schriftfarbe, Hintergrundfarbe



- fett, kursiv, durchgestrichen, unterstrichen (einfach, doppelt, gepunktet)
- hochgestellt, tiefgestellt
- Zeichenzwischenräume
- Textausrichtung (linksbündig, rechtsbündig, zentriert, Blocksatz)
- Kasten um den Text
- Zeilenumbruch, Zeilenzwischenraum
- Einrückungen, Tabulatoren, vorhergehende/nachfolgende Leerzeichen

Links, Rasterbilder oder Grafiken werden nicht mit übertragen.

- Eine Textbedeutung (Text, Pflichtattribut, nicht multipel), die die inhaltliche Bedeutung und somit die zeichnerische Ausgestaltung festlegt. Für die Bedeutungen besteht derzeit noch keine vordefinierte Liste.
- Breite und Höhe (Float, eindeutige Pflichtattribute). Die Angaben entsprechen der Bounding Box vor Anwendung des Drehwinkels.
- Die Einheit (Text) bezeichnet die Einheit von Breite und Höhe. Der Wert entspricht entweder "Welt" oder "mm". Fehlt die Angabe, so wird "Welt" angenommen. Breite und Höhe sind bei "Welt" in derselben Einheit wie die zugehörige `direct_position` (siehe Geometrie-Schema des OKSTRA) angegeben. Bei "mm" handelt es sich um Kartenmillimeter.
- Drehwinkel des Textes von  $-\pi$  bis  $+\pi$ . 0 bedeutet keine Drehung. Die Drehung setzt am Punkt zum Beschriftungsobjekt an (nicht am Bezugspunkt). Siehe hierzu die Festlegungen im OKSTRA-Teilschema zum Entwurf (Dokument D0008-1000).
- Textneigung des Textes im Bogenmaß. Die Textneigung wird relativ zur Senkrechten der Basislinie eines Textes definiert. Bei einem ungedrehten Text ist die Bezugsrichtung gegen Norden. Übliche Werte für eine Textneigung (nach DIN) sind  $\pm 15^\circ$  (Altgrad), es sind aber auch, je nach Anwendung, abweichende Neigungen zulässig. Die Textneigung impliziert, dass die Schriftart keine Neigung beinhaltet. Für das Winkelsystem gilt hier das gleiche wie für die Drehwinkel. 0 bedeutet keine Drehung.
- Schrift (Text) beschreibt den Namen des zu verwendenden TrueType-Fonts. Fehlt die Angabe, so ist der ISO-Font der RAS-Verm zu verwenden.
- Ein Flag das angibt, ob ein Kasten um den Text gesetzt werden soll.
- Ein Flag das angibt, ob eine Fahne verwendet wird. Mögliche Werte: ohne, gerade, geknickt. Fahnen werden vom Bezugspunkt zum Einfügepunkt dargestellt.
- Vertikale Ausrichtung des Textes relativ zur Textposition. Mögliche Werte: oberhalb, unterhalb, zentriert. Eine vertikale Ausrichtung "oberhalb" beispielsweise bedeutet also, dass der Text oberhalb der angegebenen Textposition abgebildet wird.
- Horizontale Ausrichtung des Textes relativ zur Textposition. Mögliche Werte: links, rechts, zentriert.



### 3 EXPRESS-Schema

Im folgenden ist das EXPRESS-Schema für die allgemeinen Geometrieobjekte angegeben. Nach der Abstimmung wird dieses als eigenständiges Schema in das Referenzschema des OKSTRA integriert.

```
SCHEMA Allgemeine_Geometrieobjekte;

REFERENCE FROM Entwurf (allgemeine_Eigenschaften);

REFERENCE FROM Geometrieschema (Punktobjekt_Modell,Linienobjekt_Modell,
                                Flaechenobjekt_Modell);

REFERENCE FROM Allgemeine_Objekte (Groesse,Winkel);

ENTITY Eigensch_allgem_Geometrieobj
ABSTRACT SUPERTYPE OF (ONEOF(allgemeines_Punktobjekt,allgemeines_Linienobjekt,
                              allgemeines_Flaechenobjekt,ALKIS_Objekt_Punkt,
                              ALKIS_Objekt_Linie,ALKIS_Objekt_Flaeche));
--- Attribute :
    Fachbedeutung           : OPTIONAL SET[1:?] OF STRING;
    Bezeichnung             : OPTIONAL STRING;
    Informationstext        : OPTIONAL SET[1:?] OF STRING;
--- Relationen :
END_ENTITY;

ENTITY allgemeines_Punktobjekt
SUBTYPE OF (Punktobjekt_Modell,Eigensch_allgem_Geometrieobj);
--- Attribute :
--- Relationen :
    hat_Symbol              : OPTIONAL SET[1:?] OF Symbol;
INVERSE
    in_Dreieck             : SET[0:?] OF Dreieck FOR hat_Punkte;
END_ENTITY;

ENTITY Symbol
SUBTYPE OF (Punktobjekt_Modell);
--- Attribute :
    Symbolbedeutung        : STRING; -- RAS-Verm-Namen als Schlüsseltab?
    Breite                 : Groesse;
    Hoehe                  : Groesse;
    Einheit                 : Einheit;
    Drehwinkel             : Winkel;
--- Relationen :
INVERSE
    mit_Bezug_auf_allg_Punktobjekt: SET[0:1] OF allgemeines_Punktobjekt
                                      FOR hat_Symbol;
END_ENTITY;

ENTITY Einheit;
    (* KEY_NAME Kennung *)
    Kennung                 : INTEGER;
    Langtext                : STRING(4);
UNIQUE
    Kennung_eindeutig      : Kennung;
END_ENTITY;

(* SQL :

INSERT INTO Kennung VALUES (0,'Welt')
INSERT INTO Kennung VALUES (1,'mm')

END_SQL
```



\*)

```
ENTITY allgemeines_Linienobjekt
SUBTYPE OF (Linienobjekt_Modell,Eigensch_allgem_Geometrieobj);
--- Attribute :
--- Relationen :
INVERSE
  in_Boeschung          : SET[0:?] OF Boeschungsschraffen
                        FOR hat_Linie;
END_ENTITY;

ENTITY Boeschungsschraffen;
--- Attribute :
  Boeschungsbedeutung   : STRING;
--- Relationen :
  hat_Linie             : SET[1:?] OF allgemeines_Linienobjekt;
END_ENTITY;

ENTITY allgemeines_Flaechenobjekt
SUBTYPE OF (Flaechenobjekt_Modell,Eigensch_allgem_Geometrieobj);
--- Attribute :
--- Relationen :
END_ENTITY;

ENTITY ALKIS_Objekt_Punkt
SUBTYPE OF (Punktobjekt_Modell,Eigensch_allgem_Geometrieobj);
--- Attribute :
--- Relationen :
  hat_ALKIS_Identifikator : ALKIS_Identifikator;
END_ENTITY;

ENTITY ALKIS_Objekt_Linie
SUBTYPE OF (Linienobjekt_Modell,Eigensch_allgem_Geometrieobj);
--- Attribute :
--- Relationen :
  hat_ALKIS_Identifikator : ALKIS_Identifikator;
END_ENTITY;

ENTITY ALKIS_Objekt_Flaeche
SUBTYPE OF (Flaechenobjekt_Modell,Eigensch_allgem_Geometrieobj);
--- Attribute :
--- Relationen :
  hat_ALKIS_Identifikator : ALKIS_Identifikator;
END_ENTITY;

ENTITY ALKIS_Identifikator;
  (* KONZEPTUELL J *)
--- Attribute :
  Identifikator          : STRING;
--- Relationen :
INVERSE
  von_ALKIS_Objekt_Punkt : SET[0:?] OF ALKIS_Objekt_Punkt
                        FOR hat_ALKIS_Identifikator;
  von_ALKIS_Objekt_Linie : SET[0:?] OF ALKIS_Objekt_Linie
                        FOR hat_ALKIS_Identifikator;
  von_ALKIS_Objekt_Flaeche : SET[0:?] OF ALKIS_Objekt_Flaeche
                        FOR hat_ALKIS_Identifikator;
END_ENTITY;

ENTITY DGM;
--- Attribute :
--- Relationen :
  hat_Dreiecke          : OPTIONAL SET[1:?] OF Dreieck;
  hat_allgemeine_Eigenschaften : allgemeine_Eigenschaften;
END_ENTITY;

ENTITY Dreieck;
--- Attribute :
  Dreiecksbedeutung     : STRING;
```



```
--- Relationen :
    hat_Punkte                : LIST[3:3] OF allgemeines_Punktobjekt;
    hat_Seiteneigenschaften   : LIST[3:3] OF Seiteneigenschaft;
INVERSE
    in_DGM                    : DGM FOR hat_Dreiecke;
END_ENTITY;
```

```
ENTITY Seiteneigenschaft;
    (* KEY_NAME Kennung *)
    Kennung                   : INTEGER;
    Langtext                   : STRING(13);
UNIQUE
    Kennung_eindeutig         : Kennung;
END_ENTITY;
```

```
(* SQL :

INSERT INTO Seiteneigenschaft VALUES (1,'normale Seite')
INSERT INTO Seiteneigenschaft VALUES (2,'Bruchlinie')
INSERT INTO Seiteneigenschaft VALUES (3,'Formlinie')

END_SQL
*)
```

-- Werte vollständig?

```
ENTITY Beschriftung
SUBTYPE OF (Punktobjekt_Modell);
--- Attribute :
    Text                       : RTF;
    Textbedeutung              : STRING;
    Breite                     : Groesse;
    Hoehe                      : Groesse;
    Einheit                    : OPTIONAL Einheit;
    Drehwinkel                 : OPTIONAL Winkel;
    Textneigung                : OPTIONAL Winkel;
    Schriftart                 : OPTIONAL STRING;
    Kasten                     : OPTIONAL BOOLEAN;
    Fahne                      : OPTIONAL Fahne;
    vertikale_Ausrichtung      : OPTIONAL vertikale_Ausrichtung;
    horizontale_Ausrichtung    : OPTIONAL horizontale_Ausrichtung;
--- Relationen :
    zu_Punktobjekt_Modell      : OPTIONAL Punktobjekt_Modell;
END_ENTITY;
```

```
TYPE RTF = STRING;
END_TYPE;
```

```
ENTITY Fahne;
    (* KEY_NAME Kennung *)
    Kennung                   : INTEGER;
    Langtext                   : STRING(8);
UNIQUE
    Kennung_eindeutig         : Kennung;
END_ENTITY;
```

```
(* SQL :

INSERT INTO Fahne VALUES (0,'ohne')
INSERT INTO Fahne VALUES (1,'gerade')
INSERT INTO Fahne VALUES (2,'geknickt')

END_SQL
*)
```

```
ENTITY vertikale_Ausrichtung;
    (* KEY_NAME Kennung *)
    Kennung                   : INTEGER;
    Langtext                   : STRING(9);
```



```
UNIQUE
  Kennung_eindeutig          : Kennung;
END_ENTITY;

(* SQL :

INSERT INTO vertikale_Ausrichtung VALUES (0,'zentriert')
INSERT INTO vertikale_Ausrichtung VALUES (1,'oberhalb')
INSERT INTO vertikale_Ausrichtung VALUES (2,'unterhalb')

  END_SQL
*)

ENTITY horizontale_Ausrichtung;
  (* KEY_NAME Kennung *)
  Kennung                    : INTEGER;
  Langtext                   : STRING(9);
UNIQUE
  Kennung_eindeutig          : Kennung;
END_ENTITY;

(* SQL :

INSERT INTO horizontale_Ausrichtung VALUES (0,'zentriert')
INSERT INTO horizontale_Ausrichtung VALUES (1,'links')
INSERT INTO horizontale_Ausrichtung VALUES (2,'rechts')

  END_SQL
*)

END_SCHEMA; -- Allgemeine_Geometrieobjekte
```

### 3.1 Auszüge anderer Schemata

Im folgenden sind Auszüge aus anderen EXPRESS-Schemata des OKSTRA angegeben. Diese sind auf die Anforderungen der allgemeinen Geometrieobjekte zugeschnitten und entsprechen nicht in allen Teilen den gültigen OKSTRA-Schemata. Diese Angaben sind also nur informativ zu sehen. Ferner sind die zu erwartenden Änderungen in diesen Schemata aufgrund der Integration der allgemeinen Geometrieobjekte hier bereits integriert.

Als weitere Auswirkung ist zu erwarten, dass das neu modellierte Objekt DGM das bisherige DGM in der Modellierung des Fachbereichs Entwurf ersetzt.

```
SCHEMA Entwurf;

(*
Auszug zur Verwendung im Schema Allgemeine_Geometrieobjekte
*)

REFERENCE FROM Allgemeine_Geometrieobjekte (DGM);

ENTITY allgemeine_Eigenschaften;
--- Attribute :
  Bezeichnung                : STRING;
  fachliche_Bedeutung        : STRING;
  Informationstext           : OPTIONAL STRING;
--- Relationen :
INVERSE
  zu_DGM                    : SET[0:1] OF DGM
                             FOR hat_allgemeine_Eigenschaften;
END_ENTITY;
```



```
END_SCHEMA; -- Entwurf

SCHEMA Geometrieschema;

(*
Auszug zur Verwendung im Schema Allgemeine_Geometrieobjekte
*)

REFERENCE FROM allgemeine_Geometrieobjekte (allgemeines_Punktobjekt,Symbol,
      allgemeines_Linienobjekt,allgemeines_Flaechenobjekt,Beschriftung,
      ALKIS_Objekt_Punkt,ALKIS_Objekt_Linie,ALKIS_Objekt_Flaeche);

REFERENCE FROM Allgemeine_Objekte (Winkel);

-- europaeische Vornorm prENV 12762:
REFERENCE FROM direct_positioning_schema (direct_position);

ENTITY Punktobjekt_Modell
ABSTRACT SUPERTYPE OF (ONEOF(allgemeines_Punktobjekt,Symbol,Beschriftung,
      ALKIS_Objekt_Punkt));
--- Attribute :
--- Relationen :
      dargestellt_von_Punkt          : OPTIONAL SET [1:?] OF Punkt;
INVERSE
      zu_Beschriftung              : SET[0:?] OF Beschriftung
      FOR zu_Punktobjekt_Modell;
END_ENTITY;

ENTITY Linienobjekt_Modell
ABSTRACT SUPERTYPE OF (allgemeines_Linienobjekt,ALKIS_Objekt_Linie);
--- Attribute :
--- Relationen :
      dargestellt_von_Linie          : OPTIONAL SET [1:?] OF Linie;
END_ENTITY;

ENTITY Flaechenobjekt_Modell
ABSTRACT SUPERTYPE OF (allgemeines_Flaechenobjekt,ALKIS_Objekt_Flaeche);
--- Attribute :
--- Relationen :
      dargestellt_von_Flaeche        : OPTIONAL SET [1:?] OF Flaeche;
END_ENTITY;

ENTITY Raumbezugsart
ABSTRACT SUPERTYPE OF (ONEOF(Punkt,Linie,Flaech));
--- Attribute :
--- Relationen :
END_ENTITY;

ENTITY Punkt
SUPERTYPE OF (gerechneter_Punkt)
SUBTYPE OF (Raumbezugsart);
--- Attribute :
--- Relationen :
      hat_Lage                      : direct_position;
INVERSE
      stellt_Punktobjekt_dar        : SET [0:?] OF Punktobjekt_Modell
      FOR dargestellt_von_Punkt;
      Beginn_von_Linienelement     : SET [0:?] OF Linienelement_3D
      FOR beginnt_bei_Punkt;
      Ende_von_Linienelement       : SET [0:?] OF Linienelement_3D
      FOR endet_bei_Punkt;
      Zwischenpunkt_von_Kreisbogen  : SET[0:?] OF Kreisbogen
      FOR hat_Zwischenpunkt;
      Stuetzpunkt_in_Spline         : SET [0:?] OF Linienelement_Spline
      FOR hat_Stuetzpunkte;
END_ENTITY;
```



```
ENTITY gerechneter_Punkt
SUBTYPE OF (Punkt);
--- Attribute :
--- Relationen :
END_ENTITY;

ENTITY Linie
SUBTYPE OF (Raumbezugsart);
(* BEMERKUNG Reihenfolge anpassen *)
(* BEMERKUNG Redundanz beseitigen *)
--- Attribute :
--- Relationen :
  besteht_aus_Linienelementen : LIST [1:?] OF Linienelement_3D;
  beschreibt_Linien_auf_Flelem  : OPTIONAL SET [1:?] OF Linie_Flaechenelement;
INVERSE
  stellt_Linienobjekt_dar      : SET [0:?] OF Linienobjekt_Modell
                                FOR dargestellt_von_Linie;
END_ENTITY;

ENTITY Linienelement_3D
ABSTRACT SUPERTYPE OF (ONEOF(Linienelement_Spline,gerades_Linienelement,
                              Kreisbogen));
--- Attribute :
--- Relationen :
  beginnt_bei_Punkt           : Punkt;
  endet_bei_Punkt             : Punkt;
INVERSE
  zu_Linie                    : Linie FOR besteht_aus_Linienelementen;
END_ENTITY;

ENTITY Linienelement_Spline
SUBTYPE OF (Linienelement_3D);
(* BEMERKUNG Reihenfolge anpassen *)
--- Attribute :
  Kruemmung_am_Anfang         : OPTIONAL Real;
  Kruemmung_am_Ende           : OPTIONAL Real;
  Tangente_am_Anfang          : OPTIONAL Winkel;
  Tangente_am_Ende            : OPTIONAL Winkel;
--- Relationen :
  hat_Stuetzpunkte            : OPTIONAL LIST [1:?] OF Punkt;
END_ENTITY;

ENTITY gerades_Linienelement
SUBTYPE OF (Linienelement_3D);
--- Attribute :
--- Relationen :
END_ENTITY;

ENTITY Kreisbogen
SUBTYPE OF (Linienelement_3D);
--- Attribute :
--- Relationen :
  hat_Zwischenpunkt          : Punkt;
END_ENTITY;

ENTITY Linie_Flaechenelement;
--- Attribute :
--- Relationen :
  bei_Linienfunktion          : STRING;
INVERSE
  Linie                        : Linie FOR beschreibt_Linien_auf_Flelem;
  Flaechenelement             : Flaechenelement
                                FOR beschrieben_durch_Linien;
END_ENTITY;

ENTITY Flaechе
ABSTRACT SUPERTYPE OF (ONEOF(Flaechenelement,komplexe_Flaechе))
SUBTYPE OF (Raumbezugsart);
--- Attribute :
```



```
--- Relationen :
INVERSE
  stellt_Flaechenobjekt_dar      : SET [0:?] OF Flaechenobjekt_Modell
                                  FOR dargestellt_von_Flaeche;
  erste_Flaeche_in              : SET [0:?] OF komplexe_Flaeche
                                  FOR hat_erste_Flaeche;
  zweite_Flaeche_in            : SET [0:?] OF komplexe_Flaeche
                                  FOR hat_zweite_Flaeche;
END_ENTITY;

ENTITY Flaechenelement
SUBTYPE OF (Flaeche);
--- Attribute :
--- Relationen :
  beschrieben_durch_Linien      : Linie_Flaechenelement;
END_ENTITY;

ENTITY komplexe_Flaeche
SUBTYPE OF (Flaeche);
--- Attribute :
--- Relationen :
  hat_erste_Flaeche            : Flaeche;
  hat_zweite_Flaeche          : Flaeche;
END_ENTITY;

END_SCHEMA; -- Geometrieschema

SCHEMA direct_positioning_schema;

REFERENCE FROM Geometrieschema (Punkt);

ENTITY direct_position;
  (* BEMERKUNG SQL anpassen *)
--- Attribute :
  in_pos_ref_system            : OPTIONAL STRING(30);
  Koordinate                   : LIST[2:3] OF REAL;
  in_coordinate_system         : OPTIONAL STRING(30);
--- Relationen :
INVERSE
  von_Punkt                    : Punkt FOR hat_Lage;
END_ENTITY;

END_SCHEMA; -- direct_positioning_schema

SCHEMA Allgemeine_Objekte;

(*
Auszug zur Verwendung im Schema Allgemeine_Geometrieobjekte
*)

TYPE Groesse = REAL;
WHERE
  Groesse_nicht_negativ       : SELF >= 0;
END_TYPE;

TYPE Winkel = REAL;
-- zwischen -Pi und Pi
END_TYPE;

END_SCHEMA; -- Allgemeine_Objekte
```



## 4 SQL-Schema

Im folgenden wird das aus dem oben gegebenen EXPRESS-Schema für die allgemeinen Geometrieobjekte sowie den Auszügen anderer EXPRESS-Schemata des OKSTRA abgeleitete SQL-Schema dargestellt.

```
CREATE SCHEMA Allgemeine_Geometrieobjekte

CREATE TABLE Eigensch_allgem_Geometrieobj (
  Eigensch_allgem_Geometrieobj_ID
    INTEGER NOT NULL PRIMARY KEY,
  Bezeichnung
    VARCHAR(255)
)

CREATE TABLE Eigensch_allgem_Geometrieobj__Fachbedeutung (
  Eigensch_allgem_Geometrieobj
    INTEGER NOT NULL REFERENCES Eigensch_allgem_Geometrieobj,
  Fachbedeutung
    VARCHAR(255) NOT NULL
)

CREATE TABLE Eigensch_allgem_Geometrieobj__Informationstext (
  Eigensch_allgem_Geometrieobj
    INTEGER NOT NULL REFERENCES Eigensch_allgem_Geometrieobj,
  Informationstext
    VARCHAR(255) NOT NULL
)

CREATE TABLE allgemeines_Punktobjekt (
  allgemeines_Punktobjekt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (allgemeines_Punktobjekt_ID) REFERENCES
  Geometrieschema.Punktobjekt_Modell,
  FOREIGN KEY (allgemeines_Punktobjekt_ID) REFERENCES
  Eigensch_allgem_Geometrieobj
)

CREATE TABLE Symbol (
  Symbol_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Symbol_ID) REFERENCES Geometrieschema.Punktobjekt_Modell,
  Symbolbedeutung
    VARCHAR(255) NOT NULL,
  Breite
    REAL NOT NULL, -- Groesse
  CHECK ( Breite >= 0 ),
  Hoehe
    REAL NOT NULL, -- Groesse
  CHECK ( Hoehe >= 0 ),
  Einheit
    INTEGER NOT NULL REFERENCES Einheit,
  Drehwinkel
    REAL NOT NULL, -- Winkel
  mit_Bezug_auf_allg_Punktobjekt
    INTEGER REFERENCES allgemeines_Punktobjekt
)

CREATE TABLE Einheit (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
```



```
        VARCHAR(4) NOT NULL
    )

INSERT INTO Kennung VALUES (0,'Welt')
INSERT INTO Kennung VALUES (1,'mm')

CREATE TABLE allgemeines_Linienobjekt (
    allgemeines_Linienobjekt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (allgemeines_Linienobjekt_ID) REFERENCES
Geometrieschema.Linienobjekt_Modell,
    FOREIGN KEY (allgemeines_Linienobjekt_ID) REFERENCES
Eigensch_allgem_Geometrieobj
)

CREATE TABLE Boeschungsschraffen (
    Boeschungsschraffen_ID
        INTEGER NOT NULL PRIMARY KEY,
    Boeschungsbedeutung
        VARCHAR(255) NOT NULL
)

CREATE TABLE Boeschungsschraffen__hat_Linie (
    Boeschungsschraffen
        INTEGER NOT NULL REFERENCES Boeschungsschraffen,
    allgemeines_Linienobjekt
        INTEGER NOT NULL REFERENCES allgemeines_Linienobjekt
)

CREATE TABLE allgemeines_Flaechenobjekt (
    allgemeines_Flaechenobjekt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (allgemeines_Flaechenobjekt_ID) REFERENCES
Geometrieschema.Flaechenobjekt_Modell,
    FOREIGN KEY (allgemeines_Flaechenobjekt_ID) REFERENCES
Eigensch_allgem_Geometrieobj
)

CREATE TABLE ALKIS_Objekt_Punkt (
    ALKIS_Objekt_Punkt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (ALKIS_Objekt_Punkt_ID) REFERENCES
Geometrieschema.Punktobjekt_Modell,
    FOREIGN KEY (ALKIS_Objekt_Punkt_ID) REFERENCES Eigensch_allgem_Geometrieobj,
    Identifikator
        VARCHAR(255) NOT NULL
)

CREATE TABLE ALKIS_Objekt_Linie (
    ALKIS_Objekt_Linie_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (ALKIS_Objekt_Linie_ID) REFERENCES
Geometrieschema.Linienobjekt_Modell,
    FOREIGN KEY (ALKIS_Objekt_Linie_ID) REFERENCES Eigensch_allgem_Geometrieobj,
    Identifikator
        VARCHAR(255) NOT NULL
)

CREATE TABLE ALKIS_Objekt_Flaeche (
    ALKIS_Objekt_Flaeche_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (ALKIS_Objekt_Flaeche_ID) REFERENCES
Geometrieschema.Flaechenobjekt_Modell,
    FOREIGN KEY (ALKIS_Objekt_Flaeche_ID) REFERENCES
Eigensch_allgem_Geometrieobj,
    Identifikator
        VARCHAR(255) NOT NULL
)
```



```
CREATE TABLE DGM (
  DGM_ID
    INTEGER NOT NULL PRIMARY KEY,
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES Entwurf.allgemeine_Eigenschaften
)

CREATE TABLE Dreieck (
  Dreieck_ID
    INTEGER NOT NULL PRIMARY KEY,
  Dreiecksbedeutung
    VARCHAR(255) NOT NULL,
  in_DGM
    INTEGER NOT NULL REFERENCES DGM
)

CREATE TABLE Dreieck__hat_Punkte (
  Dreieck
    INTEGER NOT NULL REFERENCES Dreieck,
  allgemeines_Punktobjekt
    INTEGER NOT NULL REFERENCES allgemeines_Punktobjekt,
  folgt_auf_allgemeines_Punktobjekt
    INTEGER REFERENCES allgemeines_Punktobjekt
)

CREATE TABLE Dreieck__hat_Seiteneigenschaften (
  Dreieck
    INTEGER NOT NULL REFERENCES Dreieck,
  Seiteneigenschaft
    INTEGER NOT NULL REFERENCES Seiteneigenschaft,
  folgt_auf_Seiteneigenschaft
    INTEGER REFERENCES Seiteneigenschaft
)

CREATE TABLE Seiteneigenschaft (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(13) NOT NULL
)

INSERT INTO Seiteneigenschaft VALUES (1,'normale Seite')
INSERT INTO Seiteneigenschaft VALUES (2,'Bruchlinie')
INSERT INTO Seiteneigenschaft VALUES (3,'Formlinie')

CREATE TABLE Beschriftung (
  Beschriftung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Beschriftung_ID) REFERENCES Geometrieschema.Punktobjekt_Modell,
  Text
    VARCHAR(255) NOT NULL, -- RTF
  Textbedeutung
    VARCHAR(255) NOT NULL,
  Breite
    REAL NOT NULL, -- Groesse
  CHECK ( Breite >= 0 ),
  Hoehe
    REAL NOT NULL, -- Groesse
  CHECK ( Hoehe >= 0 ),
  Einheit
    INTEGER REFERENCES Einheit,
  Drehwinkel
    REAL, -- Winkel
  Textneigung
    REAL, -- Winkel
  Schriftart
    VARCHAR(255),
  Kasten
    INTEGER REFERENCES Logical_Types.Boolean,
```



```
Fahne
  INTEGER REFERENCES Fahne,
vertikale_Ausrichtung
  INTEGER REFERENCES vertikale_Ausrichtung,
horizontale_Ausrichtung
  INTEGER REFERENCES horizontale_Ausrichtung,
zu_Punktobjekt_Modell
  INTEGER REFERENCES Geometrieschema.Punktobjekt_Modell
)

CREATE TABLE Fahne (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(8) NOT NULL
)

INSERT INTO Fahne VALUES (0,'ohne')
INSERT INTO Fahne VALUES (1,'gerade')
INSERT INTO Fahne VALUES (2,'geknickt')

CREATE TABLE vertikale_Ausrichtung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(9) NOT NULL
)

INSERT INTO vertikale_Ausrichtung VALUES (0,'zentriert')
INSERT INTO vertikale_Ausrichtung VALUES (1,'oberhalb')
INSERT INTO vertikale_Ausrichtung VALUES (2,'unterhalb')

CREATE TABLE horizontale_Ausrichtung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(9) NOT NULL
)

INSERT INTO horizontale_Ausrichtung VALUES (0,'zentriert')
INSERT INTO horizontale_Ausrichtung VALUES (1,'links')
INSERT INTO horizontale_Ausrichtung VALUES (2,'rechts')

CREATE SCHEMA Entwurf

CREATE TABLE allgemeine_Eigenschaften (
  allgemeine_Eigenschaften_ID
    INTEGER NOT NULL PRIMARY KEY,
  Bezeichnung
    VARCHAR(255) NOT NULL,
  fachliche_Bedeutung
    VARCHAR(255) NOT NULL,
  Informationstext
    VARCHAR(255)
)

CREATE SCHEMA Geometrieschema

CREATE TABLE Punktobjekt_Modell (
  Punktobjekt_Modell_ID
    INTEGER NOT NULL PRIMARY KEY
)
```



```
CREATE TABLE Punktobjekt_Modell__dargestellt_von_Punkt (
  Punktobjekt_Modell
    INTEGER NOT NULL REFERENCES Punktobjekt_Modell,
  Punkt
    INTEGER NOT NULL REFERENCES Punkt
)

CREATE TABLE Linienobjekt_Modell (
  Linienobjekt_Modell_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Linienobjekt_Modell__dargestellt_von_Linie (
  Linienobjekt_Modell
    INTEGER NOT NULL REFERENCES Linienobjekt_Modell,
  Linie
    INTEGER NOT NULL REFERENCES Linie
)

CREATE TABLE Flaechenobjekt_Modell (
  Flaechenobjekt_Modell_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Flaechenobjekt_Modell__dargestellt_von_Flaeche (
  Flaechenobjekt_Modell
    INTEGER NOT NULL REFERENCES Flaechenobjekt_Modell,
  Flaeche
    INTEGER NOT NULL REFERENCES Flaeche
)

CREATE TABLE Raumbezugsart (
  Raumbezugsart_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Punkt (
  Punkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Punkt_ID) REFERENCES Raumbezugsart,
  hat_Lage
    INTEGER NOT NULL REFERENCES direct_positioning_schema.direct_position
)

CREATE TABLE gerechneter_Punkt (
  gerechneter_Punkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (gerechneter_Punkt_ID) REFERENCES Punkt
)

CREATE TABLE Linie (
  Linie_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Linie_ID) REFERENCES Raumbezugsart
)

CREATE TABLE Linie__besteht_aus_Linienelementen (
  Linie
    INTEGER NOT NULL REFERENCES Linie,
  Linienelement_3D
    INTEGER NOT NULL REFERENCES Linienelement_3D,
  folgt_auf_Linienelement_3D
    INTEGER REFERENCES Linienelement_3D
)

CREATE TABLE Linienelement_3D (
  Linienelement_3D_ID
    INTEGER NOT NULL PRIMARY KEY,
  beginnt_bei_Punkt
```



```
        INTEGER NOT NULL REFERENCES Punkt,
    endet_bei_Punkt
        INTEGER NOT NULL REFERENCES Punkt,
    zu_Linie
        INTEGER NOT NULL REFERENCES Linie
)

CREATE TABLE Linienelement_Spline (
    Linienelement_Spline_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Linienelement_Spline_ID) REFERENCES Linienelement_3D,
    Kruemmung_am_Anfang
        REAL,
    Kruemmung_am_Ende
        REAL,
    Tangente_am_Anfang
        REAL, -- Winkel
    Tangente_am_Ende
        REAL -- Winkel
)

CREATE TABLE Linienelement_Spline__hat_Stuetzpunkte (
    Linienelement_Spline
        INTEGER NOT NULL REFERENCES Linienelement_Spline,
    Punkt
        INTEGER NOT NULL REFERENCES Punkt,
    folgt_auf_Punkt
        INTEGER REFERENCES Punkt
)

CREATE TABLE gerades_Linienelement (
    gerades_Linienelement_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (gerades_Linienelement_ID) REFERENCES Linienelement_3D
)

CREATE TABLE Kreisbogen (
    Kreisbogen_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Kreisbogen_ID) REFERENCES Linienelement_3D,
    hat_Zwischenpunkt
        INTEGER NOT NULL REFERENCES Punkt
)

CREATE TABLE Linie_Flaechenelement (
    Linie_Flaechenelement_ID
        INTEGER NOT NULL PRIMARY KEY,
    bei_Linienfunktion
        VARCHAR(255) NOT NULL,
    Linie
        INTEGER NOT NULL REFERENCES Linie
)

CREATE TABLE Flaechе (
    Flaechе_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Flaechе_ID) REFERENCES Raumbezugsart
)

CREATE TABLE Flaechenelement (
    Flaechenelement_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Flaechenelement_ID) REFERENCES Flaechе,
    beschrieben_durch_Linien
        INTEGER NOT NULL REFERENCES Linie_Flaechenelement
)

CREATE TABLE komplexe_Flaechе (
    komplexe_Flaechе_ID
```



```
        INTEGER NOT NULL PRIMARY KEY,  
        FOREIGN KEY (komplexe_Flaeche_ID) REFERENCES Flaechen,  
        hat_erste_Flaeche  
        INTEGER NOT NULL REFERENCES Flaechen,  
        hat_zweite_Flaeche  
        INTEGER NOT NULL REFERENCES Flaechen  
    )
```

```
CREATE SCHEMA direct_positioning_schema
```

```
CREATE TABLE direct_position (  
    direct_position_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    in_pos_ref_system  
        VARCHAR(30),  
    Koordinate_1  
        REAL NOT NULL,  
    Koordinate_2  
        REAL NOT NULL,  
    Koordinate_3  
        REAL,  
    in_coordinate_system  
        VARCHAR(30)  
)
```

```
CREATE SCHEMA Logical_Types
```

```
CREATE TABLE Boolean (  
    Boolean_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    Boolean  
        VARCHAR (5)  
)
```

```
INSERT INTO Boolean VALUES ( 0 , 'FALSE' )  
INSERT INTO Boolean VALUES ( 1 , 'TRUE' )
```



	<b>Objektkatalog für das Straßen- und Verkehrswesen</b> <b>Modellierung der allgemeinen Geometrieobjekte</b>	<b>Seite: 27 von 27</b> <b>Name: N0011</b> <b>Stand: 26.09.2000</b>
--	---	---

## 5.1 Erläuterungen

Im oben abgebildeten Geometrie-Schema ist gegenüber dem aktuell gültigen Geometrie-Schema des OKSTRA als zusätzliches Linienelement der *Kreisbogen* ergänzt worden. Ein Kreisbogen wird durch drei (verschiedene) Punkte beschrieben. Durch die Festlegung, dass der zusätzliche Punkt auf dem Kreisbogen zwischen den begrenzenden Punkten des 3D-Linienelements liegen muss, ist der Kreisbogen eindeutig beschrieben.

Zu beachten ist hierbei, dass es sich bei den Entwurfselementen um 2D-Elemente handelt. Es sind Projektionen der Raumkurven in den Grundriss. Eine echte 3D-Darstellung ist nach wie vor Gegenstand der wissenschaftlichen Forschung. Aufgrund der Orientierung des OKSTRA an den bestehenden Regelwerken kann die Integration einer vollen 3D-Geometrie erst nach Fortschreibung der entsprechenden Regelwerke erfolgen.

Bei den Splines handelt es sich um kubische Splines. Fehlt die Krümmungsrandbedingung, so kann diese vom Zielsystem frei gewählt werden. Fehlen Krümmung und Tangentenwinkel, so handelt es sich um einen natürlichen Spline. Bei geschlossenen Splines wird der zusätzliche Schalter gesetzt. In diesen Fällen sind die Randbedingungen am Anfang und am Ende vom Zielsystem zur Deckung zu bringen. Schließt der Spline tangential an seine umgebenden Linienelemente an, so ist der entsprechende Schalter gesetzt. Splines gemäß OKSTRA interpolieren ihre Stützpunkte.