Objektkatalog für das Straßen- und Verkehrswesen

Vorschlag zur Einbindung des dynamischen Querprofils in den OKSTRA®

Version:	1.0
Datum:	05.05.2003
Status:	Vorschlag
Dateiname:	N0050
Pfad:	n.a.
Verantwortlich:	Dietmar König

OKSTRA-Pflegestelle

interactive instruments GmbH
Trierer Straße 70-72
53115 Bonn

Herr Dietmar König
Tel. 0228 91410 76
Fax 0228 91410 90
Email koenig@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

0 Allgemeines

0.1 Inhaltsverzeichnis

0.2 Änderungen

Name	Datum	Kapitel	Bemerkungen	Bearbeiter
N0050	05.05.2003	alle	Dokument erstellt auf Basis der Experten-Besprechung zur Einbindung des dynamischen Querprofils in den OKSTRA®	Dietmar König

0.3 Bezüge

Name	Bemerkungen
Zwischenbericht zur geometrischen Modellierung	Dokument erhältlich auf den OKSTRA®-Webseiten unter N0046.pdf: http://www.okstra.de/docs/n0046.pdf

0.4 Bearbeitungsvermerke

Der Vorschlag wurde in einer Experten-Besprechung mit Auftragnehmern des Forschungsvorhabens zum dynamischen Querprofil erarbeitet. An dieser Experten-Besprechung haben teilgenommen (in alphabetischer Reihenfolge):

Name	Institution
Herr Feser	AKG Software Consulting GmbH. Ballrechten-Dottingen

Herr König	interactive instruments GmbH, Bonn
Herr Kornbichler	München
Herr Reiter	RIB Bausoftware GmbH, Stuttgart

1 Zweck des Dokuments

1.1 Leserkreis

Das Dokument richtet sich an alle Experten im Bereich des dynamischen Querprofils bzw. der Neubaudaten im OKSTRA®.

1.2 Kernaussagen des Inhalts

Dieses Dokument enthält Vorschläge zur Einbindung des dynamischen Querprofils in den OKSTRA®. Grundlage sind die entsprechenden Ergebnisse des Forschungsvorhabens zum dynamischen Querprofil. Teile der bisherigen Modellierung der Neubaudaten im OKSTRA® werden dabei überarbeitet.

Die Modellierungsvorschläge wurden in Zusammenarbeit mit den oben genannten Experten erstellt.

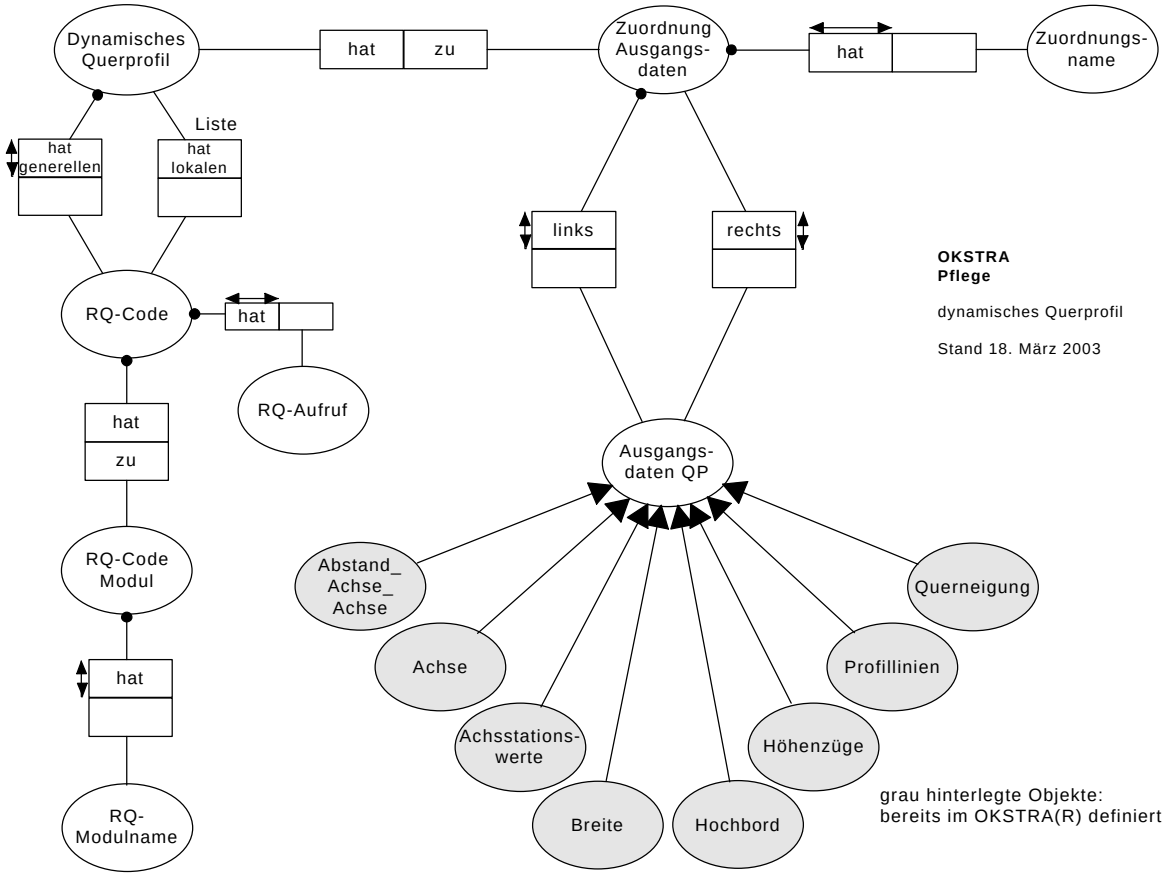
Für weitere Erläuterungen wird auf das Forschungsvorhaben zum dynamischen Querprofil verwiesen. Auf den OKSTRA®-Webseiten ist der Zwischenbericht zur geometrischen Modellierung (des dynamischen Querprofils) erhältlich: http://www.okstra.de/docs/n0046.pdf.

Die Modellierungsvorschläge werden zunächst als NIAM-Diagramme dokumentiert und erläutert. Anschließend wird die Referenzmodellierung in EXPRESS gegeben.

2 Vorschläge zur Modellierung

2.1 Dynamisches Querprofil

Der aktuelle OKSTRA® enthält nur einen Platzhalter "Bildungsgesetze" mit zugehörigen "Ausgangsdaten" für das dynamische Querprofil. Bildungsgesetze und Ausgangsdaten werden durch die nachfolgende Modellierung des dynamischen Querprofils ersetzt:



Jedes dynamische Querprofil hat genau einen generellen RQ-Code. Dieser kann durch eine Liste von beliebig vielen lokalen RQ-Codes ergänzt werden. In diesem Fall wird zunächst der generelle RQ-Code angewendet und anschließend die lokalen RQ-Codes in der angegebenen Reihenfolge auf die jeweiligen Zwischenergebnisse.

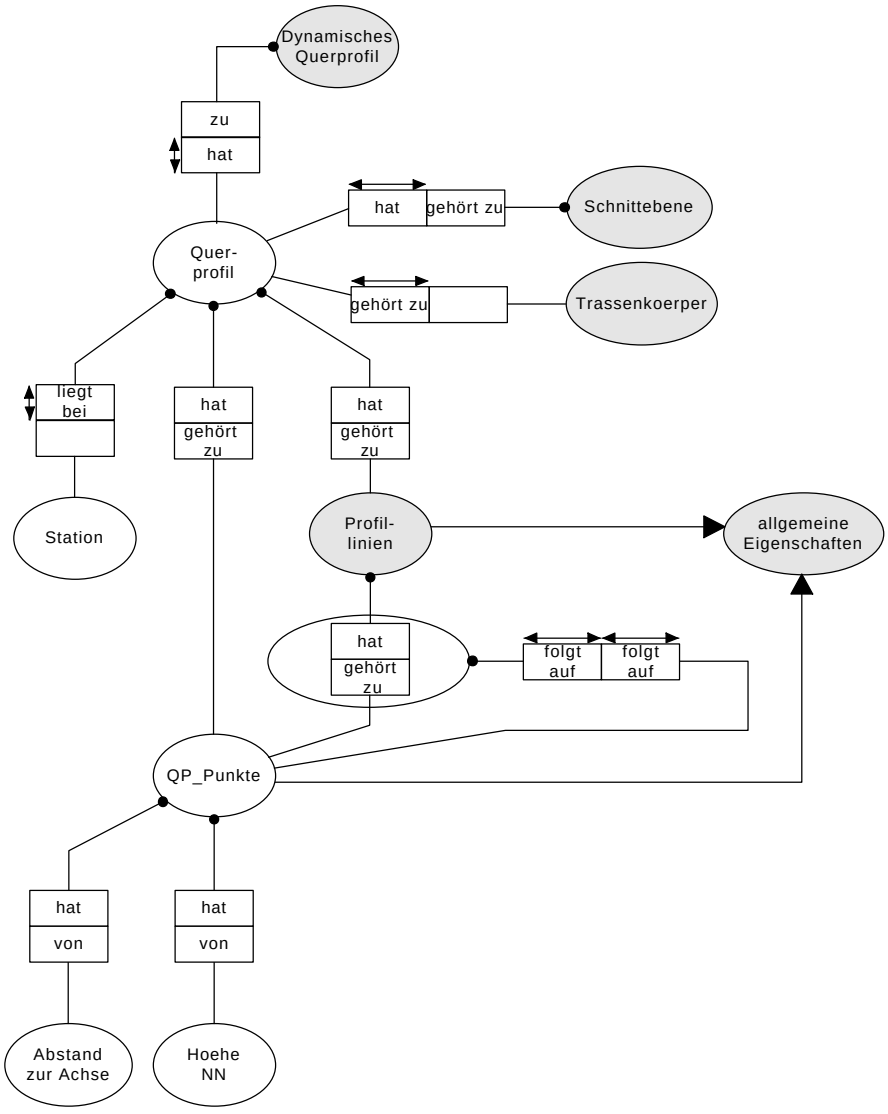
Die Ausgangsdaten des dynamischen Querprofils werden als abstrakter Supertype von acht bereits modellierten Fachobjekten aus den OKSTRA® Neubaudaten definiert. Jede Zuordnung hat ein Ausgangsdatum für die linke Seite. Für die rechte Seite kann ein anderes Ausgangsdatum angegeben werden. Wird kein zweites Ausgangsdatum angegeben, so gilt das Ausgangsdatum der linken Seite für beide Seiten.

Die Wirksamkeit bzw. Anwendbarkeit des dynamischen Querprofils bei veränderten Ausgangsdaten hängt von der Qualität des dynamischen Querprofils ab, z.B. ob gewisse Randbedingungen bei der Erstellung des dynamischen Querprofils berücksichtigt worden sind oder nicht.

Der RQ-Code wird in den OKSTRA®-Daten grundsätzlich zeichengenau als STRING eingebettet, also einschließlich aller Leerzeichen, Tabulatoren, Zeilenwechsel etc.

2.2 Querprofil

Jedes dynamische Querprofil wird einem oder mehreren (statischen) Querprofilen zugeordnet. Von diesem bezieht das dynamische Querprofil seine Station. Die Anbindung ist in folgendem überarbeiteten NIAM-Diagramm zum statischen Querprofil gegeben:

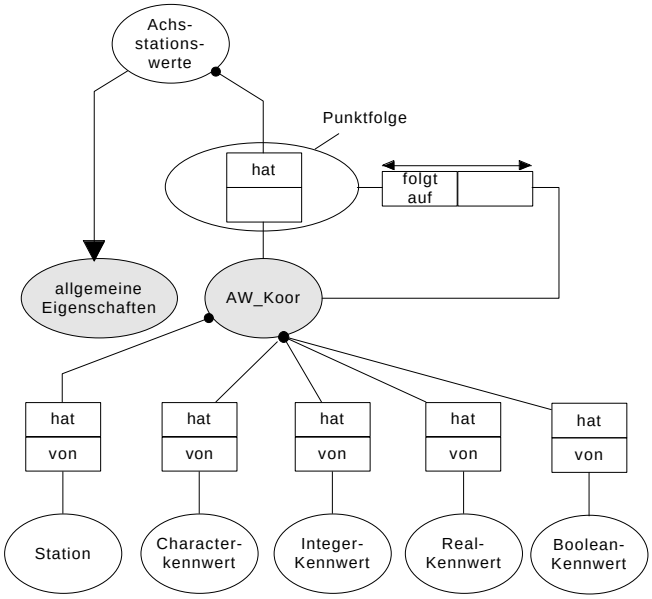


Zwischen Stationen mit gleichem RQ-Code wird bewusst keine Aussage über die Anwendbarkeit des RQ-Codes getroffen. Hier muss die Anwendung bzw. der Anwender entscheiden.

Die Bezeichnung aus den allgemeinen Eigenschaften der QP-Punkte wird als Punktname verwendet. Er muss eindeutig pro Profillinie sein. In der Fachbedeutung kann z.B. die Herkunft beschrieben werden. Dies wird noch im Rahmen des Forschungsvorhabens zum dynamischen Querprofil genauer beschrieben.

2.3 Achsstationswerte

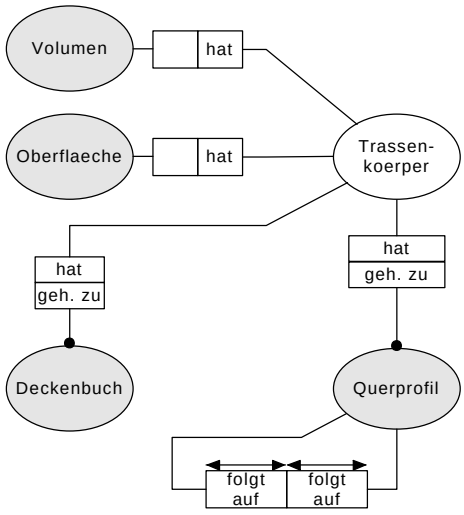
Für die Achsstationswerte können im Objekt AW_Koor auch boolesche Werte angegeben werden. Es ergibt sich folgendes geändertes NIAM-Diagramm:



Mit Achsstationswerten werden alle Informationen abgelegt, die die Bildungsgesetze steuern und abschnittsweise konstant sind. Mit einem Achsstationswerte-Objekt werden entweder Character-, Real-, Boolean- oder Integer-Werte beschrieben. Es wird jeweils genau ein Wert angegeben.

2.4 Trassenkörper

Aus der Modellierung des dynamische Querprofils ergibt sich folgendes geändertes NIAM-Diagramm für den Trassenkörper:



3 EXPRESS

Das folgende EXPRESS-Schema stellt den Vorschlag für die Einbindung des dynamischen Querprofils in das Schema Entwurf des OKSTRA® dar. Dieser Vorschlag ersetzt nach der Abstimmung das bestehende Schema Entwurf:

```
SCHEMA Entwurf;

(*
Vorschlag zur Einbindung
des dynamischen Querprofils
*)

REFERENCE FROM Strassennetz (Netzknoten_abstrakt,Knotenpunktsform);

REFERENCE FROM Ingenieurbauwerke (Bauwerkseinzelheiten);

REFERENCE FROM Ausstattung (Beschilderung_Lichtsignalanl,Fahrbahnmarkierungen,
                            Schutz_und_Leiteinrichtungen);

REFERENCE FROM Geometrieschema (Punktobjekt_Modell,Linienobjekt_Modell);

REFERENCE FROM Allgemeine_Objekte
(Meter,Kilometer,Groesse,Winkel,Stundenkilometer,Prozent);

REFERENCE FROM Allgemeine_Geometrieobjekte (Dreieck,allgemeines_Punktobjekt,
                  allgemeines_Linienobjekt,allgemeines_Flaechenobjekt);

REFERENCE FROM Kataster
(ALKIS_Objekt_Punkt,ALKIS_Objekt_Linie,ALKIS_Objekt_Flaeche);

ENTITY Trasse
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
--- Relationen :
   hat_Netzknoten               : OPTIONAL Netzknoten_abstrakt; (* O *)
   hat_Achse                    : OPTIONAL SET [1:?] OF Achse;
   hat_Entwurfsparameter        : OPTIONAL SET [1:?] OF Entwurfsparameter;
   hat_Laengsschnitt            : OPTIONAL SET [1:?] OF Laengsschnitt;
   hat_Trassenkoerper           : OPTIONAL SET [1:?] OF Trassenkoerper;
   hat_Sichtweiten              : OPTIONAL SET [1:?] OF Sichtweiten;
END_ENTITY;

ENTITY Achse
SUBTYPE OF (allgemeine_Eigenschaften,Ausgangsdaten_QP);
   (* BEMERKUNG Reihenfolge anpassen *)
--- Attribute :
--- Relationen :
   hat_Achselement              : LIST [1:?] OF Achselement;
INVERSE
```

```
   gehoert_zu_Trasse             : Trasse FOR hat_Achse;
   hat_Laengsschnitt             : SET [0:1] OF Laengsschnitt
                                               FOR gehoert_zu_Achse;
   hat_Trassenkoerper            : SET [0:1] OF Trassenkoerper
                                               FOR gehoert_zu_Achse;
   hat_Sichtweiten               : SET [0:?] OF Sichtweiten
                                               FOR gehoert_zu_Achse;
   von_Kreuzungs_o_Einmuend_plang: SET [0:?] OF Kreuzungs_o_Einmuendungsplang
                                               FOR hat_Achse;
   von_Abstand_Achse_Achse       : SET [0:?] OF Abstand_Achse_Achse
                                               FOR hat_zweite_Achse;
END_ENTITY;

ENTITY Achselement
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   Elementtyp                    : Achselementtyp;
   Anfangsstation_rechnerisch    : Meter;
   Verwaltungsstation_Betriebskm : Kilometer;
   Laenge                        : Meter;
   Richtung                      : Winkel;
   Parameter                     : Groesse;
   Radius_zu_Beginn              : Meter;
   Radius_am_Ende                : Meter;
--- Relationen :
   beginnt_bei_Achshauptpunkt    : Achshauptpunkt;
   endet_bei_Achshauptpunkt      : Achshauptpunkt;
INVERSE
   gehoert_zu_Achse              : SET [0:1] OF Achse FOR hat_Achselement;
END_ENTITY;

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

(* SQL :

INSERT INTO Achselementtyp VALUES (1,'Gerade')
INSERT INTO Achselementtyp VALUES (2,'Kreisbogen, tangential')
INSERT INTO Achselementtyp VALUES (12,'Klothoide')

   END_SQL
*)

ENTITY Achshauptpunkt
SUBTYPE OF (Punktobjekt_Modell);
--- Attribute :
--- Relationen :
```

```
INVERSE
   Beginn_von_Achselement        : SET [0:?] OF Achselement
                                             FOR beginnt_bei_Achshauptpunkt;
   Ende_von_Achselement          : SET [0:?] OF Achselement
                                             FOR endet_bei_Achshauptpunkt;
   zu_Kreuzungs_o_Einmuendplanung: SET [0:?] OF Kreuzungs_o_Einmuendungsplang
                                             FOR hat_Achshauptpunkt;
END_ENTITY;

ENTITY Kreuzungs_o_Einmuendungsplang
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   Netzknotennummer              : STRING(7);
   Knotenpunktsform              : Knotenpunktsform;
--- Relationen :
   hat_Achse                     : SET [1:?] OF Achse;
   hat_DGM                       : OPTIONAL DGM;
   hat_Achshauptpunkt            : Achshauptpunkt;
END_ENTITY;

ENTITY DGM
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
--- Relationen :
   hat_Dreiecke                  : OPTIONAL SET[1:?] OF Dreieck;
INVERSE
   gehoert_zu_Kreuz_o_Einmplang  : SET [0:?] OF Kreuzungs_o_Einmuendungsplang
                                             FOR hat_DGM;
END_ENTITY;

ENTITY Laengsschnitt  -- Der Laengsschnitt beschreibt einen Hoehenverlauf.
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
--- Relationen :
   gehoert_zu_Achse              : Achse;
   hat_kreuzende_Bauw_o_baul_Anl : OPTIONAL SET [1:?]
                                           OF kreuzende_Bauwerke_o_baul_Anl;
   hat_Gelaendehorizonte         : OPTIONAL SET [1:?] OF Gelaendehorizonte;
   hat_Gradiente                 : OPTIONAL SET [1:?] OF Gradiente;
INVERSE
   gehoert_zu_Trasse             : Trasse FOR hat_Laengsschnitt;
END_ENTITY;

ENTITY Gelaendehorizonte
SUBTYPE OF (Laengsschnittlinie);
--- Attribute :
--- Relationen :
INVERSE
   gehoert_zu_Laengsschnitt      : Laengsschnitt FOR hat_Gelaendehorizonte;
END_ENTITY;

ENTITY Gradiente
```

```
SUBTYPE OF (Laengsschnittlinie);
--- Attribute :
--- Relationen :
INVERSE
   gehoert_zu_Laengsschnitt      : Laengsschnitt FOR hat_Gradiente;
   zu_Hoehe_Gradiente            : SET [0:?] OF Hoehe_Gradiente
                                         FOR hat_Gradiente;
END_ENTITY;

ENTITY kreuzende_Bauwerke_o_baul_Anl
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
--- Relationen :
   hat_Tunnel_Kanal_Strasse_etc  : OPTIONAL Tunnel_Kanal_Strasse_Durchlass;
   hat_Schnittgeometrie          : SET [1:?] OF Schnittgeometrie;
INVERSE
   gehoert_zu_Laengsschnitt      : SET [0:?] OF Laengsschnitt
                                         FOR hat_kreuzende_Bauw_o_baul_Anl;
END_ENTITY;

ENTITY Tunnel_Kanal_Strasse_Durchlass;
--- Attribute :
--- Relationen :
INVERSE
   von_kreuz_Bauwerken_o_baul_Anl: SET [0:?] OF kreuzende_Bauwerke_o_baul_Anl
                                         FOR hat_Tunnel_Kanal_Strasse_etc;
   hat_Schnittgeometrie          : SET [0:?] OF Schnittgeometrie
                                      FOR abgel_von_Tunnel_Kanal_Str_etc;
END_ENTITY;

ENTITY Schnittgeometrie;
--- Attribute :
   Schnittstation                : Meter;
   Schnitthoehe                  : Meter;
   Schnittwinkel_horizontal      : Winkel;
   Laengsneigung_kreuzend_Bauwerk: Winkel;
--- Relationen :
   abgel_von_Tunnel_Kanal_Str_etc: Tunnel_Kanal_Strasse_Durchlass;
   hat_Schnittpolygone           : SET [1:?] OF Schnittpolygone;
   hat_Mindestabstandspolygone   : OPTIONAL SET [1:?]
                                         OF Mindestabstandspolygon;
   hat_Maximalabstandspolygone   : OPTIONAL SET [1:?]
                                         OF Maximalabstandspolygon;
INVERSE
   geh_zu_kreuz_Bauw_o_baul_Anl  : SET [0:?] OF kreuzende_Bauwerke_o_baul_Anl
                                         FOR hat_Schnittgeometrie;
END_ENTITY;

ENTITY Schnittpolygone
SUBTYPE OF (Polygon);
--- Attribute :
--- Relationen :
```

```
INVERSE
   gehoert_zu_Schnittgeometrie   : SET [1:?] OF Schnittgeometrie
                                              FOR hat_Schnittpolygone;
END_ENTITY;

ENTITY Mindestabstandspolygon
SUBTYPE OF (Polygon);
--- Attribute :
--- Relationen :
INVERSE
   gehoert_zu_Schnittgeometrie   : SET [1:?] OF Schnittgeometrie
                                              FOR hat_Mindestabstandspolygone;
END_ENTITY;

ENTITY Maximalabstandspolygon
SUBTYPE OF (Polygon);
--- Attribute :
--- Relationen :
INVERSE
   gehoert_zu_Schnittgeometrie   : SET [1:?] OF Schnittgeometrie
                                              FOR hat_Maximalabstandspolygone;
END_ENTITY;

ENTITY Polygon
ABSTRACT SUPERTYPE OF (ONEOF(Schnittpolygone,Maximalabstandspolygon,
                             Mindestabstandspolygon));
--- Attribute :
--- Relationen :
   hat_QP_Punkte                 : LIST [1:?] OF QP_Punkte;
END_ENTITY;

ENTITY Laengsschnittlinie
SUPERTYPE OF (ONEOF(Gelaendehorizonte,Gradiente))
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   abs_Abstand                   : OPTIONAL REAL(16);
--- Relationen :
   hat_LS_Koor                   : LIST [1:?] OF LS_Koor;
   hat_Breite                    : OPTIONAL SET [1:?] OF Breite;
WHERE
   Bezug_eindeutig               : NOT EXISTS(abs_Abstand) OR NOT
EXISTS(hat_Breite);
END_ENTITY;

ENTITY LS_Koor
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   Station                       : Meter;
   Hoehe                         : Meter;
--- Relationen :
   folgt_auf_LS_Koor             : OPTIONAL Punktfolge;
INVERSE
```

```
   gehoert_zu_Laengsschnittlinie : SET [0:?] OF Laengsschnittlinie
                                          FOR hat_LS_Koor;
END_ENTITY;

ENTITY Punktfolge;
--- Attribute :
--- Relationen :
   hat_Tangente_Gerade           : Tangente_Gerade;
INVERSE
   LS_Koor_Nachfolger            : LS_Koor FOR folgt_auf_LS_Koor;
END_ENTITY;

ENTITY Tangentenfolge;
--- Attribute :
--- Relationen :
   hat_Ausrundung                : OPTIONAL Ausrundung;
INVERSE
   Tangente_Gerade_Nachfolger    : Tangente_Gerade FOR
                                          folgt_auf_Tangente_Gerade;
END_ENTITY;

ENTITY Ausrundung;
--- Attribute :
   Ausrundungstyp                : Ausrundungstyp;
   Ausrundungsparameter          : Groesse;
--- Relationen :
INVERSE
   gehoert_zu_Tangentenfolge     : SET [0:1] OF Tangentenfolge
                                          FOR hat_Ausrundung;
END_ENTITY;

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

(* SQL :

INSERT INTO Ausrundungstyp VALUES (13,'Parabel 2. Grades')
INSERT INTO Ausrundungstyp VALUES (14,'Parabel 3. Grades mit langem Teil '+
 'vor TS')
INSERT INTO Ausrundungstyp VALUES (15,'Parabel 3. Grades mit langem Teil '+
 'hinter TS')

   END_SQL
*)

ENTITY Tangente_Gerade;
--- Attribute :
```

```
--- Relationen :
   folgt_auf_Tangente_Gerade     : OPTIONAL Tangentenfolge;
INVERSE
   gehoert_zu_Punktfolge         : SET [0:1] OF Punktfolge
                                             FOR hat_Tangente_Gerade;
END_ENTITY;

ENTITY Volumen_aus_QP
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   Station_1                     : Meter;
   Station_2                     : Meter;
--- Relationen :
   wird_begrenzt_von_Profillinie : Profillinien;
INVERSE
   von_Trassenkoerper            : SET [0:?] OF Trassenkoerper FOR hat_Volumen;
END_ENTITY;

ENTITY Oberflaeche
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   Art_der_Oberflaeche           : INTEGER;
   Station_1                     : Meter;
   Station_2                     : Meter;
--- Relationen :
   liegt_auf_Profillinie         : Profillinien;
   beginnt_bei_QP_Punkt          : QP_Punkte;
   endet_bei_QP_Punkt            : QP_Punkte;
INVERSE
   von_Trassenkoerper            : SET [0:?] OF Trassenkoerper
                                             FOR hat_Oberflaeche;
END_ENTITY;

ENTITY Trassenkoerper;
--- Attribute :
--- Relationen :
   hat_Volumen                   : OPTIONAL SET [1:?] OF Volumen_aus_QP;
   hat_Oberflaeche               : OPTIONAL SET [1:?] OF Oberflaeche;
   hat_Deckenbuch                : OPTIONAL Deckenbuch;
   hat_Querprofil                : OPTIONAL LIST [1:?] OF Querprofil;
   gehoert_zu_Achse              : Achse;
INVERSE
   gehoert_zu_Trasse             : Trasse FOR hat_Trassenkoerper;
END_ENTITY;

ENTITY dynamisches_Querprofil;
--- Attribute:
--- Relationen:
   hat_Zuordnung_Ausgangsdaten   : OPTIONAL SET [1:?] OF
Zuordnung_Ausgangsdaten;
   hat_generellen_RQ_Code        : RQ_Code;
   hat_lokalen_RQ_Code           : OPTIONAL LIST [1:?] OF RQ_Code;
```

```
   zu_Querprofil                : Querprofil;
END_ENTITY;

ENTITY RQ_Code;
--- Attribute:
   RQ_Aufruf                    : STRING;
--- Relationen:
   hat_RQ_Code_Modul            : SET [1:?] OF RQ_Code_Modul;
INVERSE
   ist_genereller_RQ_Code       : SET [0:?] OF dynamisches_Querprofil
                                            FOR hat_generellen_RQ_Code;
   ist_lokaler_RQ_Code          : SET [0:?] OF dynamisches_Querprofil
                                            FOR hat_lokalen_RQ_Code;
END_ENTITY;

ENTITY RQ_Code_Modul;
--- Attribute:
   RQ_Modulname                 : STRING;
   Code                         : STRING; -- zeichengenauer RQ-Code
--- Relationen:
INVERSE
   zu_RQ_Code                   : SET [0:?] OF RQ_Code FOR hat_RQ_Code_Modul;
END_ENTITY;

ENTITY Zuordnung_Ausgangsdaten;
--- Attribute:
   Zuordnungsname               : STRING;
--- Relationen:
   hat_Ausgangsdaten_links_beide : Ausgangsdaten_QP;
   hat_Ausgangsdaten_rechts     : OPTIONAL Ausgangsdaten_QP;
INVERSE
   zu_dynamischem_Querprofil    : SET [0:?] OF dynamisches_Querprofil
                                            FOR hat_Zuordnung_Ausgangsdaten;
END_ENTITY;

ENTITY Ausgangsdaten_QP
ABSTRACT SUPERTYPE OF (ONEOF(Abstand_Achse_Achse,Achse,Achsstationswerte,

Breite,Hochbord,Hoehenzuege,Profillinien,Querneigung);
--- Attribute:
--- Relationen:
INVERSE
   in_Zuordnung_links_beide     : SET [0:?] OF Zuordnung_Ausgangsdaten
                                            FOR hat_Ausgangsdaten_links_beide;
   in_Zuordnung_rechts          : SET [0:?] OF Zuordnung_Ausgangsdaten
                                            FOR hat_Ausgangsdaten_rechts;
END_ENTITY;

ENTITY Querprofil;
--- Attribute :
   Station                      : Meter;
--- Relationen :
```

```
   hat_Schnittebene              : OPTIONAL Schnittebene;
   hat_Profillinien              : SET [1:?] OF Profillinien;
   hat_QP_Punkte                 : OPTIONAL SET [1:?] OF QP_Punkte;
INVERSE
   gehoert_zu_Trassenkoerper     : SET [0:1] OF Trassenkoerper
                                             FOR hat_Querprofil;
   hat_dynamisches_Querprofil    : SET [0:?] OF dynamisches_Querprofil
                                             FOR zu_Querprofil;
END_ENTITY;

ENTITY QP_Punkte
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   Abstand_zur_Achse             : REAL(16);
   Hoehe_NN                      : REAL(16);
--- Relationen :
INVERSE
   gehoert_zu_Profillinien       : SET [0:?] OF Profillinien FOR hat_QP_Punkte;
   gehoert_zu_Querprofil         : SET [0:?] OF Querprofil FOR hat_QP_Punkte;
   Beginn_von_Oberflaeche        : SET [0:?] OF Oberflaeche
                                             FOR beginnt_bei_QP_Punkt;
   Ende_von_Oberflaeche          : SET [0:?] OF Oberflaeche
                                             FOR endet_bei_QP_Punkt;
   Beginn_von_Spur_aus_Querprof  : SET [0:?] OF Spur_aus_Querprofilen
                                             FOR beginnt_bei_QP_Punkt;
   Ende_von_Spur_aus_Querprof    : SET [0:?] OF Spur_aus_Querprofilen
                                             FOR endet_bei_QP_Punkt;
   gehoert_zu_Polygon            : SET [0:?] OF Polygon FOR hat_QP_Punkte;
END_ENTITY;

ENTITY Profillinien
SUBTYPE OF (allgemeine_Eigenschaften,Ausgangsdaten_QP);
--- Attribute :
--- Relationen :
   hat_QP_Punkte                 : LIST [1:?] OF QP_Punkte;
INVERSE
   gehoert_zu_Querprofil         : SET [0:?] OF Querprofil
                                             FOR hat_Profillinien;
   gehoert_zu_Oberflaeche        : SET [0:?] OF Oberflaeche
                                             FOR liegt_auf_Profillinie;
   begrenzt_Volumen_aus_QP       : SET [0:4] OF Volumen_aus_QP
                                             FOR wird_begrenzt_von_Profillinie;
   von_Spur_aus_Querprofilen     : SET [0:?] OF Spur_aus_Querprofilen
                                             FOR liegt_auf_Profillinie;
END_ENTITY;

ENTITY Schnittebene;
--- Attribute :
--- Relationen :
   hat_SNT_Punkte                : OPTIONAL LIST [1:?] OF SNT_Punkt;
INVERSE
   gehoert_zu_Querprofil         : SET [1:?] OF Querprofil
```

```
                                          FOR hat_Schnittebene;
END_ENTITY;

ENTITY SNT_Punkt;
--- Attribute :
   Abstand                      : Meter;
   Richtungsaenderung           : REAL(16);
   korrespondierende_Achse      : OPTIONAL STRING(255);
   Naeherungsstation            : OPTIONAL Meter;
--- Relationen :
INVERSE
   gehoert_zu_Schnittebene      : SET [1:?] OF Schnittebene
                                          FOR hat_SNT_Punkte;
END_ENTITY;

ENTITY Deckenbuch
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   Station_1                    : Meter;
   Station_2                    : Meter;
--- Relationen :
   hat_Spur_aus_Ausgangsdaten   : OPTIONAL LIST [1:?]
                                          OF Spur_aus_Ausgangsdaten;
   hat_Spur_aus_Querprofilen    : OPTIONAL LIST [1:?]
                                          OF Spur_aus_Querprofilen;
INVERSE
   gehoert_zu_Trassenkoerper    : SET [1:?] OF Trassenkoerper
                                          FOR hat_Deckenbuch;
WHERE
   Ausgangsdaten_oder_Querprofile: EXISTS(hat_Spur_aus_Ausgangsdaten)
                                    XOR EXISTS(hat_Spur_aus_Querprofilen);
END_ENTITY;

ENTITY Spur_aus_Ausgangsdaten
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
--- Relationen :
   hat_Querneigung              : OPTIONAL Querneigung;
   hat_Breite                   : Breite;
   hat_Hoehenzuege              : OPTIONAL Hoehenzuege;
   hat_Hochbord                 : OPTIONAL Hochbord;
INVERSE
   von_Deckenbuch               : SET [1:?] OF Deckenbuch
                                       FOR hat_Spur_aus_Ausgangsdaten;
   zu_BR_Punkt                  : SET [0:?] OF BR_Punkt
                                       FOR bez_auf_Spur_aus_Ausgangsdaten;
   ist_Fahrspur_zu              : SET [0:?] OF Sichtweiten
                                       FOR hat_Fahrspur;
   ist_Gegenspur_zu             : SET [0:?] OF Sichtweiten
                                       FOR hat_Gegenspur;
END_ENTITY;
```

```
ENTITY Spur_aus_Querprofilen
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
--- Relationen :
   liegt_auf_Profillinie         : Profillinien;
   beginnt_bei_QP_Punkt          : QP_Punkte;
   endet_bei_QP_Punkt            : QP_Punkte;
INVERSE
   von_Deckenbuch                : SET [1:?] OF Deckenbuch
                                       FOR hat_Spur_aus_Querprofilen;
END_ENTITY;

ENTITY Breite
SUBTYPE OF (allgemeine_Eigenschaften,Ausgangsdaten_QP);
--- Attribute :
--- Relationen :
   hat_BR_Punkt                  : LIST [1:?] OF BR_Punkt;
INVERSE
   von_Spur_aus_Ausgangsdaten    : SET [0:?] OF Spur_aus_Ausgangsdaten
                                       FOR hat_Breite;
   gehoert_zu_Laengsschnittlinie : SET [0:?] OF Laengsschnittlinie
                                       FOR hat_Breite;
END_ENTITY;

ENTITY BR_Punktfolge;
--- Attribute :
--- Relationen :
   hat_Aufweitg_Verbreit_Verbind : Aufweitung_Verbreit_Verbind;
INVERSE
   vor_BR_Punkt                  : BR_Punkt FOR folgt_auf_BR_Punkt;
END_ENTITY;

ENTITY BR_Punkt;
--- Attribute :
   Breite                        : OPTIONAL Meter;
   Station                       : Meter;
--- Relationen :
   hat_Abstand_Achse_Achse       : OPTIONAL Abstand_Achse_Achse;
   hat_Abstand_Achse_Linie       : OPTIONAL Abstand_Achse_Linie;
   folgt_auf_BR_Punkt            : OPTIONAL BR_Punktfolge;
   bez_auf_Spur_aus_Ausgangsdaten: OPTIONAL Spur_aus_Ausgangsdaten;
INVERSE
   gehoert_zu_Breite             : Breite FOR hat_BR_Punkt;
WHERE
   genau_eine_Abstandsangabe     : (EXISTS(Breite)
                                    AND NOT EXISTS(hat_Abstand_Achse_Achse)
                                    AND NOT EXISTS(hat_Abstand_Achse_Linie))
                                   OR (EXISTS(hat_Abstand_Achse_Achse)
                                    AND NOT EXISTS(Breite)
                                    AND NOT EXISTS(hat_Abstand_Achse_Linie))
                                   OR (EXISTS(hat_Abstand_Achse_Linie)
                                    AND NOT EXISTS(Breite)
```

```
                                    AND NOT EXISTS(hat_Abstand_Achse_Achse));
   (* BEDINGUNG (Breite IS NOT NULL AND hat_Abstand_Achse_Achse IS NULL
                  AND hat_Abstand_Achse_Linie IS NULL)
             OR (Breite IS NULL AND hat_Abstand_Achse_Achse IS NOT NULL
                  AND hat_Abstand_Achse_Linie IS NULL)
             OR (Breite IS NULL AND hat_Abstand_Achse_Achse IS NULL
                  AND hat_Abstand_Achse_Linie IS NOT NULL) *)
END_ENTITY;

ENTITY Aufweitung_Verbreit_Verbind;
--- Attribute :
   Art_der_Verziehung            : Art_der_Verziehung;
   Tangentenlaenge               : OPTIONAL Meter;
--- Relationen :
INVERSE
   gehoert_zu_Breite_BR_Punkt    : SET [1:?] OF BR_Punktfolge
                                             FOR hat_Aufweitg_Verbreit_Verbind;
END_ENTITY;

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

(* SQL :

INSERT INTO Art_der_Verziehung VALUES (1,'Parabelfolge 2. Grades')
INSERT INTO Art_der_Verziehung VALUES (2,'Bogenfolge')
INSERT INTO Art_der_Verziehung VALUES (3,'Parabelfolge 2. Grades /
Zwischengerade')
INSERT INTO Art_der_Verziehung VALUES (4,'Gerade')
INSERT INTO Art_der_Verziehung VALUES (5,'Bezug auf Referenzobjekt')

   END_SQL
*)

ENTITY Abstand_Achse_Achse
SUBTYPE OF (Ausgangsdaten_QP,Ausgangsdaten_QP);
--- Attribute :
   Naeherungsstation_auf_zw_Achse: Meter;
--- Relationen :
   hat_Lage_der_Knicklinie       : Lage_der_Knicklinie;
   hat_zweite_Achse              : Achse;
INVERSE
   von_BR_Punkt                  : SET [0:?] OF BR_Punkt
                                             FOR hat_Abstand_Achse_Achse;
END_ENTITY;

ENTITY Abstand_Achse_Linie
```

```
SUBTYPE OF (Linienobjekt_Modell);
--- Attribute :
--- Relationen :
INVERSE
   von_BR_Punkt                 : SET [0:?] OF BR_Punkt
                                            FOR hat_Abstand_Achse_Linie;
END_ENTITY;

ENTITY Lage_der_Knicklinie;
--- Attribute :
   Typ_der_Knicklinie           : Typ_der_Knicklinie;
   Breite                       : OPTIONAL Meter;
--- Relationen :
INVERSE
   von_Abstand_Achse_Achse      : SET [0:?] OF Abstand_Achse_Achse
                                            FOR hat_Lage_der_Knicklinie;
END_ENTITY;

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

(* SQL :

INSERT INTO Typ_der_Knicklinie VALUES (1,'Parallele / Breite zur zweiten
Achse')
INSERT INTO Typ_der_Knicklinie VALUES (2,'Parallele / Breite zur Achse')
INSERT INTO Typ_der_Knicklinie VALUES (3,'Mittig mit Abstand zur Knicklinie')

   END_SQL
*)

ENTITY Hoehenzuege
SUBTYPE OF (allgemeine_Eigenschaften,Ausgangsdaten_QP);
--- Attribute :
--- Relationen :
   hat_HZ_Punkte                : LIST [1:?] OF HZ_Punkt;
INVERSE
   von_Spur_aus_Ausgangsdaten   : SET [0:?] OF Spur_aus_Ausgangsdaten
                                            FOR hat_Hoehenzuege;
END_ENTITY;

ENTITY HZ_Punkt;
--- Attribute :
   Station                      : Meter;
   Hoehe_fest                   : OPTIONAL Meter;
--- Relationen :
   hat_Hoehe_Gradiente          : OPTIONAL Hoehe_Gradiente;
```

```
   hat_Hoehe_Linie               : OPTIONAL Hoehe_Linie;
INVERSE
   gehoert_zu_Hoehenzug          : Hoehenzuege FOR hat_HZ_Punkte;
WHERE
   Bezug_eindeutig               : (EXISTS(Hoehe_fest)
                                    AND NOT EXISTS(hat_Hoehe_Gradiente)
                                    AND NOT EXISTS(hat_Hoehe_Linie))
                                    OR (EXISTS(hat_Hoehe_Gradiente)
                                    AND NOT EXISTS(Hoehe_fest)
                                    AND NOT EXISTS(hat_Hoehe_Linie))
                                    OR (EXISTS(hat_Hoehe_Linie)
                                    AND NOT EXISTS(Hoehe_fest)
                                    AND NOT EXISTS(hat_Hoehe_Gradiente));
   (* BEDINGUNG (Hoehe_fest IS NOT NULL AND hat_Hoehe_Gradiente IS NULL
                  AND hat_Hoehe_Linie IS NULL)
              OR (Hoehe_fest IS NULL AND hat_Hoehe_Gradiente IS NOT NULL
                  AND hat_Hoehe_Linie IS NULL)
              OR (Hoehe_fest IS NULL AND hat_Hoehe_Gradiente IS NULL
                  AND hat_Hoehe_Linie IS NOT NULL) *)
END_ENTITY;

ENTITY Hoehe_Gradiente;
--- Attribute :
   Naeherungsstation_Gradiente   : Meter;
--- Relationen :
   hat_Gradiente                 : Gradiente;
INVERSE
   von_HZ_Punkt                  : SET [0:?] OF HZ_Punkt
                                             FOR hat_Hoehe_Gradiente;
END_ENTITY;

ENTITY Hoehe_Linie
SUBTYPE OF (Linienobjekt_Modell);
--- Attribute :
--- Relationen :
INVERSE
   von_HZ_Punkt                  : SET [0:?] OF HZ_Punkt
                                             FOR hat_Hoehe_Linie;
END_ENTITY;

ENTITY Querneigung
SUBTYPE OF (allgemeine_Eigenschaften,Ausgangsdaten_QP);
--- Attribute :
--- Relationen :
   hat_QN_Punkte                 : LIST [1:?] OF QN_Punkt;
INVERSE
   von_Spur_aus_Ausgangsdaten    : SET [0:?] OF Spur_aus_Ausgangsdaten
                                             FOR hat_Querneigung;
END_ENTITY;

ENTITY Querneigungswechsel;
--- Attribute :
```

```
   Verziehungsform               : Verziehungsform;
--- Relationen :
INVERSE
   vor_QN_Punkt                  : QN_Punkt FOR folgt_auf_QN_Punkt;
END_ENTITY;

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

(* SQL :

INSERT INTO Verziehungsform VALUES (0,'reserviert')
INSERT INTO Verziehungsform VALUES (1,'normale Verziehung')
INSERT INTO Verziehungsform VALUES (2,'Verziehung mit Gratlinie '+
 '(Schrägverwindung)')

   END_SQL
*)

ENTITY QN_Punkt;
--- Attribute :
   Station                       : Meter;
   Querneigung                   : Prozent;
   automatische_Berechnung       : OPTIONAL BOOLEAN;
--- Relationen :
   folgt_auf_QN_Punkt            : OPTIONAL Querneigungswechsel;
INVERSE
   gehoert_zu_Querneigung        : Querneigung FOR hat_QN_Punkte;
END_ENTITY;

ENTITY Hochbord
SUBTYPE OF (allgemeine_Eigenschaften,Ausgangsdaten_QP);
--- Attribute :
--- Relationen :
   hat_HB_Punkte                 : LIST [1:?] OF HB_Punkt;
   hat_HB_Neigung                : OPTIONAL SET [1:?] OF HB_Neigung;
INVERSE
   von_Spur_aus_Ausgangsdaten    : SET [0:?] OF Spur_aus_Ausgangsdaten FOR
hat_Hochbord;
END_ENTITY;

ENTITY HB_Punkt;
--- Attribute :
   Station                       : Meter;
   Differenzhoehe                : Meter;
--- Relationen :
INVERSE
```

```
   gehoert_zu_Hochbord           : Hochbord FOR hat_HB_Punkte;
END_ENTITY;

ENTITY HB_Neigung;
--- Attribute :
   Station                       : Meter;
   Neigungswinkel                : Groesse;
--- Relationen :
INVERSE
   zu_Hochbord                   : SET [0:?] OF Hochbord FOR hat_HB_Neigung;
END_ENTITY;

ENTITY Achsstationswerte
SUBTYPE OF (allgemeine_Eigenschaften,Ausgangsdaten_QP);
--- Attribute :
--- Relationen :
   hat_AW_Koor                   : LIST [1:?] OF AW_Koor;
END_ENTITY;

ENTITY AW_Koor;
--- Attribute :
   Station                       : Meter;
   Character_Kennwert            : OPTIONAL STRING;
   Integer_Kennwert              : OPTIONAL INTEGER;
   Real_Kennwert                 : OPTIONAL REAL;
   Boolean_Kennwert              : OPTIONAL BOOLEAN;
--- Relationen :
INVERSE
   gehoert_zu_Achsstationswerten : SET [0:?] OF Achsstationswerte
                                              FOR hat_AW_Koor;
WHERE
   nur_ein_Kennwert              : (EXISTS(Character_Kennwert)
                                    AND NOT EXISTS(Integer_Kennwert)
                                    AND NOT EXISTS(Real_Kennwert)
                                    AND NOT EXISTS(Boolean_Kennwert))
                                   OR (EXISTS(Integer_Kennwert)
                                    AND NOT EXISTS(Character_Kennwert)
                                    AND NOT EXISTS(Real_Kennwert)
                                    AND NOT EXISTS(Boolean_Kennwert))
                                   OR (EXISTS(Real_Kennwert)
                                    AND NOT EXISTS(Character_Kennwert)
                                    AND NOT EXISTS(Integer_Kennwert)
                                    AND NOT EXISTS(Boolean_Kennwert))
                                   OR (EXISTS(Boolean_Kennwert)
                                    AND NOT EXISTS(Character_Kennwert)
                                    AND NOT EXISTS(Integer_Kennwert)
                                    AND NOT EXISTS(Real_Kennwert));
   (* BEDINGUNG (Character_Kennwert IS NOT NULL AND Integer_Kennwert IS NULL
                  AND Real_Kennwert IS NULL AND Boolean_Kennwert IS NULL)
             OR (Character_Kennwert IS NULL AND Integer_Kennwert IS NOT NULL
                  AND Real_Kennwert IS NULL AND Boolean_Kennwert IS NULL)
             OR (Character_Kennwert IS NULL AND Integer_Kennwert IS NULL
```

```
                AND Real_Kennwert IS NOT NULL AND Boolean_Kennwert IS NULL)
            OR (Character_Kennwert IS NULL AND Integer_Kennwert IS NULL
                AND Real_Kennwert IS NULL AND Boolean_Kennwert IS NOT NULL)
*)
END_ENTITY;

ENTITY Entwurfsparameter;
--- Attribute :
   Strassenkategorie            : OPTIONAL STRING(255);
   Verkehrsbelastung            : OPTIONAL INTEGER;
   Dokumentenverweis            : OPTIONAL STRING(255);
   Regelquerschnitt             : OPTIONAL STRING(255);
--- Relationen :
   hat_Geschwindigkeitsband     : OPTIONAL SET [1:?] OF Geschwindigkeitsband;
INVERSE
   gehoert_zu_Trasse            : Trasse FOR hat_Entwurfsparameter;
END_ENTITY;

ENTITY Geschwindigkeitsband
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
--- Relationen :
   hat_V_Koor                   : LIST [1:?] OF V_Koor;
INVERSE
   von_Entwurfsparameter        : SET [0:?] OF Entwurfsparameter
                                              FOR hat_Geschwindigkeitsband;
END_ENTITY;

ENTITY V_Koor;
--- Attribute :
   Station                      : Meter;
   Geschwindigkeit              : Stundenkilometer;
--- Relationen :
INVERSE
   gehoert_zu_Geschwindigkeitsbd : SET [0:?] OF Geschwindigkeitsband
                                              FOR hat_V_Koor;
END_ENTITY;

ENTITY Sichtweiten
SUBTYPE OF (allgemeine_Eigenschaften);
--- Attribute :
   Sichtweite                   : REAL(16);
--- Relationen :
   gehoert_zu_Achse             : Achse;
   hat_SW_Koor                  : LIST [1:?] OF SW_Koor;
   hat_Fahrspur                 : OPTIONAL Spur_aus_Ausgangsdaten;
   hat_Gegenspur                : OPTIONAL Spur_aus_Ausgangsdaten;
INVERSE
   gehoert_zu_Trasse            : Trasse FOR hat_Sichtweiten;
END_ENTITY;

ENTITY SW_Koor;
```

```
--- Attribute :
   Station                      : Meter;
   Sichtweite                   : Meter;
--- Relationen :
INVERSE
   von_Sichtweite               : SET [0:?] OF Sichtweiten FOR hat_SW_Koor;
END_ENTITY;

ENTITY allgemeine_Eigenschaften
ABSTRACT SUPERTYPE OF (ONEOF(Beschilderung_Lichtsignalanl,Fahrbahnmarkierungen,
                        Schutz_und_Leiteinrichtungen,Bauwerkseinzelheiten,
                        Trasse,Achse,Achselement,Kreuzungs_o_Einmuendungsplang,
                        Laengsschnitt,kreuzende_Bauwerke_o_baul_Anl,
                        Laengsschnittlinie,LS_Koor,Hochbord,Hoehenzuege,
                        Profillinien,QP_Punkte,Volumen_aus_QP,DGM,
                        Oberflaeche,Deckenbuch,Spur_aus_Ausgangsdaten,
                        Spur_aus_Querprofilen,Breite,Querneigung,
                        Achsstationswerte,Geschwindigkeitsband,Sichtweiten,
                        allgemeines_Punktobjekt,allgemeines_Linienobjekt,
                        allgemeines_Flaechenobjekt,ALKIS_Objekt_Punkt,
                        ALKIS_Objekt_Linie,ALKIS_Objekt_Flaeche));
--- Attribute :
   Bezeichnung                  : OPTIONAL STRING;
   fachliche_Bedeutung          : STRING;
   Informationstext             : OPTIONAL SET [1:?] OF STRING;
--- Relationen :
END_ENTITY;

END_SCHEMA; -- Entwurf
```