

Im Auftrag des  
Bundesministeriums für Verkehr,  
Bau- und Wohnungswesen

Forschungsbericht FE-Nr. 09.092 G95D

Standardisierung graphischer Daten  
im Straßen- und Verkehrswesen

Teil 2 - Realisierung

Teilbericht E: Der Datenaustausch

interactive instruments  
Gesellschaft für Softwareentwicklung mbH  
Trierer Straße 70-72  
53115 Bonn

15.10.1999



Im Auftrag des  
Bundesministeriums für Verkehr,  
Bau- und Wohnungswesen

Forschungsbericht FE-Nr. 09.092 G95D

Standardisierung graphischer Daten  
im Straßen- und Verkehrswesen

Teil 2 - Realisierung

Teilbericht E: Der Datenaustausch

von

Dipl. Phys. Clemens Portele

Dipl. Math. Dietmar König

interactive instruments  
Gesellschaft für Softwareentwicklung mbH  
Trierer Straße 70-72  
53115 Bonn

15.10.1999



## **Inhaltsverzeichnis**

<b>1</b>	<b>Das Austauschverfahren für OKSTRA-Daten</b>	<b>7</b>
<b>2</b>	<b>Beispiel: Austausch einer Netzknotenfolge</b>	<b>8</b>
2.1.1	Auszug aus dem EXPRESS-Datenschema	8
2.1.2	Darstellung einer <i>Netzknotenfolge</i>	11
<b>3</b>	<b>Aspekte zum Datenaustausch zwischen OKSTRA-Implementierungen</b>	<b>16</b>

**Inhalt dieses Dokuments:**

- Das Verfahren zum Austausch von OKSTRA-Daten: Dieser Teilbericht beschreibt beispielhaft das für den OKSTRA gewählte, genormte Datenaustauschverfahren *Clear Text Encoding*, das Teil derselben ISO-Norm (ISO 10303) ist, der auch EXPRESS entstammt.

**Zielgruppe:**

- vor allem Software-Entwickler
- fachliche Details können auch für die Experten des Straßen- und Verkehrswesens von Interesse sein

**Voraussetzungen zum Verständnis:**

- sinnvoll ist die Kenntnis von EXPRESS (ISO 10303-11) und *Clear Text Encoding* (ISO 10303-21)

## 1 Das Austauschverfahren für OKSTRA-Daten

Für EXPRESS gibt es eine genormte Methode, um Daten, die auf der Basis eines beliebigen EXPRESS-Datenschemas gegeben sind, in Textform zu repräsentieren. Diese Methode heißt **Clear Text Encoding** (im folgenden mit **CTE** bezeichnet). Sie ist Teil 21 der ISO-Norm 10303 (*Industrial Automation Systems and Integration - Product Data Representation and Exchange*), der auch EXPRESS (Teil 11) entstammt, so daß sich dieses als Standardweg anbietet und bereits in der OKSTRA-Studie vorgeschlagen wurde.

Ein großer Vorteil dieser Darstellung ist, daß die Struktur der Daten exakt der Struktur im Referenzschema des OKSTRA entspricht. Das bedeutet, daß die Austauschdaten völlig unabhängig von der speziellen Implementierung des EXPRESS-Datenschemas (beispielsweise in Form einer SQL-Datenbank) sind. Diese Lösung hebt sich daher im Hinblick auf eine Standardisierung, die mit dem OKSTRA erreicht werden soll, von allen anderen denkbaren Lösungen ab.

Ferner macht man sich durch die Verwendung einer internationalen Norm unabhängig von speziellen Software-Produkten mit denen die Austauschdaten transportiert werden. Alle für den Datenaustausch benötigten Informationen sind verfügbar, entweder als ISO-Norm oder als OKSTRA selbst. Im Fall eines speziellen Produkts ist dies nicht der Fall.

Eine Voraussetzung für die korrekte Interpretation eines CTE-Files auf Basis des OKSTRA bzw. für die Wiedergewinnung der Daten aus einer Implementierung des OKSTRA in der Struktur des EXPRESS-Datenschemas ist die Verfügbarkeit einer maschinell verarbeitbaren Form dieses Datenschemas. Ein geeignetes Programm kann damit nach den Vorgaben des Benutzers Daten aus der Implementierung des OKSTRA lesen und sie gemäß dem EXPRESS-Datenschema in CTE überführen (natürlich mit dem Wissen, wie die Implementierung aus dem Datenschema abgeleitet wurde).

## 2 Beispiel: Austausch einer Netzknotenfolge

Zur Illustration wird für eine reale Netzknotenfolge eine Darstellung in der Datenart 100 (DA100) und eine Darstellung in CTE (ISO 10303-21) angeben.

### 2.1.1 Auszug aus dem EXPRESS-Datenschema

Folgende Auszüge aus dem OKSTRA-Datenschema werden für das Beispiel verwendet:

```
ENTITY Strasse -- ASB 97, 3
SUBTYPE OF (historisches_Objekt);
--- Attribute :
--- Relationen :
    hat_Strassenbezeichnung : Strassenbezeichnung;
    hat_Strassenknoten       : SET [2:?] OF Strasse_Netzknoten; (* L *)
    hat_Abschnitt_oder_Ast   : SET [1:?] OF Abschnitt_oder_Ast;
    hat_BAB_Knotennummer    : OPTIONAL SET [1:?] OF BAB_Knotennummer;
    in_Verwaltungsbezirk    : SET [1:?] OF Verwaltungsbezirk; (* D *)
INVERSE
    enthaelt_Block          : SET [0:?] OF Block FOR auf_Strasse;
END_ENTITY;
```

```
ENTITY Strassenbezeichnung; -- ASB 97, 3.1
--- Attribute :
    Strassenklasse          : Strassenklasse;
    Strassennummer         : Strassennummer;
    Zusatzbuchstabe        : OPTIONAL Zusatzbuchstabe;
--- Relationen :
DERIVE
    Strassenname            : STRING(6) :=
                            Strassenklasse.Kennung +
                            FORMAT( Strassennummer, '4I' ) +
                            Zusatzbuchstabe_oder_Leerzeichen
                                (Zusatzbuchstabe);
INVERSE
    von_Strasse             : SET [1:?] OF Strasse
                            FOR hat_Strassenbezeichnung;
END_ENTITY;
```

```
ENTITY Strassenklasse; -- ASB 97, 3.1
(* KEY_NAME Kennung *)
(* KEY_TYP CHAR(1) *)
    Kennung                 : STRING(1);
    Langtext                : STRING;
UNIQUE
    Kennung_eindeutig       : Kennung;
END_ENTITY;

TYPE Strassennummer = INTEGER; -- ASB 97, 3.1
WHERE
    maximal_vierstellig     : { 0 < SELF <= 9999 };
    (* BEDINGUNG ( SELF > 0 ) AND ( SELF <= 9999 ) *)
END_TYPE;

TYPE Zusatzbuchstabe = STRING(1); -- ASB 97, 3.1
WHERE
    erlaubte_Buchstaben     : { 'A' <= SELF <= 'Z' };
    (* BEDINGUNG ( SELF >= 'A' ) AND ( SELF <= 'Z' ) *)
END_TYPE;

FUNCTION Zusatzbuchstabe_oder_Leerzeichen(zb:Zusatzbuchstabe) : STRING(1);
    IF EXISTS(zb) THEN
        RETURN(zb);
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht E - Datenaustausch

---

```
END_IF;  
RETURN(' ');  
END_FUNCTION;
```

```
ENTITY Strasse_Netzknoten          -- ASB 97, 3.3  
SUBTYPE OF (historisches_Objekt);  
--- Attribute :  
  Verlaufskennzeichen            : OPTIONAL Verlaufskennzeichen;  
--- Relationen :  
  hat_Netzknoten                 : SET [1:?] OF Netzknoten; (* D *)  
  hat_Vorgaenger                 : OPTIONAL SET [1:?] OF Strasse_Netzknoten;  
INVERSE  
  hat_Nachfolger                 : SET [0:?] OF Strasse_Netzknoten  
                                FOR hat_Vorgaenger;  
  von_Strasse                    : SET [1:?] OF Strasse FOR hat_Strassenknoten;  
                                (* D *)  
END_ENTITY;
```

```
ENTITY Verlaufskennzeichen;      -- ASB 97, 3.4  
  (* KEY_NAME Kennung *)  
  Kennung                        : INTEGER;  
  Langtext                       : STRING;  
UNIQUE  
  Kennung_eindeutig              : Kennung;  
END_ENTITY;
```

```
ENTITY Bundesland  
SUBTYPE OF (Verwaltungsbezirk);  
--- Attribute :  
  Kennung_Bundesland             : STRING(2) FIXED;  
--- Relationen :  
INVERSE  
  enthaelt_Regierungsbezirk      : SET [0:?] OF Regierungsbezirk  
                                FOR ist_in_Bundesland;  
END_ENTITY;  
  
ENTITY Verwaltungsbezirk        -- ASB 97, 4.2 & 9.2  
ABSTRACT SUPERTYPE OF (ONEOF(Bundesland,Regierungsbezirk,  
                             Kreis_kreisfreie_Stadt,Gemeindebezirk,Ortsteil))  
SUBTYPE OF (Bereichsobjekt_hist,ASB_Bezeichnung,Flaechenobjekt_Modell);  
  (* ARTEN J *)  
--- Attribute :  
--- Relationen :  
INVERSE  
  von_Strasse                    : SET [0:?] OF Strasse  
                                FOR in_Verwaltungsbezirk;  
  von_verwaltungstechn_Zuordnung : SET [0:?] OF verwaltungstechn_Zuordnung  
                                FOR hat_Verwaltungsbezirk;  
  von_dynam_verkehrsreg_Beschild : SET [0:?] OF verwaltungstechn_Zuordnung_dyn  
                                FOR hat_Verwaltungsbezirk;  
END_ENTITY;
```

```
ENTITY Netzknoten                -- ASB 97, 1.1  
SUBTYPE OF (Punktobjekt_Modell,historisches_Objekt);  
  (* KEY_TYP CHAR(7) *)  
--- Attribute :  
  Numerierungsbezirk             : TK25_Blattnummer;  
  Nummer                         : lfd_NK_Nummer;  
  Knotenart                      : OPTIONAL Knotenart;  
  Knotenname                     : OPTIONAL STRING(30);  
  Knotenpunktsform               : OPTIONAL Knotenpunktsform;  
  Knotenpunktsystem              : OPTIONAL Knotenpunktsystem;  
--- Relationen :  
  hat_BAB_Knotennummer           : OPTIONAL SET [1:?] OF BAB_Knotennummer;  
INVERSE
```

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht E - Datenaustausch

```
in_Strassenknoten      : SET [1:?] OF Strasse_Netzknoten
                        :           FOR hat_Netzknoten;
hat_Nullpunkt          : SET [1:?] OF Nullpunkt FOR in_Netzknoten;
Beginn_von             : SET [0:?] OF WW_Wirkungsbereiche
                        :           FOR beginnt_bei_Netzknoten;
Ende_von               : SET [0:?] OF WW_Wirkungsbereiche
                        :           FOR endet_bei_Netzknoten;
END_ENTITY;
```

```
TYPE TK25_Blattnummer = INTEGER; -- ASB 97, 1.1
WHERE
  vierstellig          : { 1 <= SELF <= 9999 };
  (* BEDINGUNG ( SELF >= 1 ) AND ( SELF <= 9999 ) *)
END_TYPE;
```

```
TYPE lfd_NK_Nummer = INTEGER; -- ASB 97, 1.1
WHERE
  dreistellig          : { 1 <= SELF <= 999 };
  (* BEDINGUNG ( SELF >= 1 ) AND ( SELF <= 999 ) *)
END_TYPE;
```

```
ENTITY Knotenart; -- ASB 97, 1.1.4
  (* KEY_NAME Kennung *)
  Kennung              : INTEGER;
  Langtext             : STRING;
UNIQUE
  Kennung_eindeutig   : Kennung;
END_ENTITY;
```

```
ENTITY Knotenpunktsform; -- ASB 97, 1.1.6
  (* KEY_NAME Kennung *)
  Kennung              : INTEGER;
  Langtext             : STRING;
UNIQUE
  Kennung_eindeutig   : Kennung;
END_ENTITY;
```

```
ENTITY Knotenpunktsystem; -- ASB 97, 1.1.7
  (* KEY_NAME Kennung *)
  Kennung              : INTEGER;
  Langtext             : STRING;
UNIQUE
  Kennung_eindeutig   : Kennung;
END_ENTITY;
```

```
ENTITY Punktobjekt_Modell
ABSTRACT SUPERTYPE OF (ONEOF(Netzknoten,Nullpunkt));
  (* FOREIGN_KEY N *)
  (* VERERBEN SUBID *)
--- Attribute :
--- Relationen :
  dargestellt_von_Knoten      : OPTIONAL SET [1:?] OF Knoten;
  dargestellt_von_Punkt       : OPTIONAL SET [1:?] OF Punkt;
END_ENTITY;
```

```
ENTITY historisches_Objekt
ABSTRACT SUPERTYPE OF (ONEOF(Strassenbezeichnung,Strasse,Strasse_Netzknoten,
  Netzknoten, ...));
  (* VERERBEN SUBID *)
  (* ARTEN J *)
  (* MIX_IN gueltig_von,gueltig_bis *)
--- Attribute :
  gueltig_von                : OPTIONAL Datum;
  gueltig_bis                : OPTIONAL Datum;
--- Relationen :
  erzeugt_von_Ereignis       : Ereignis;
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht E - Datenaustausch

---

```
geloescht_von_Ereignis      : OPTIONAL Ereignis;
hat_Vorgaenger_hist_Objekt  : OPTIONAL historisches_Objekt;
INVERSE
  hat_Nachfolger_hist_Objekt : SET [0:1] OF historisches_Objekt
                                FOR hat_Vorgaenger_hist_Objekt;
WHERE
  Objektfolge_konsistent   : Objektfolge_konsistent(SELF);
END_ENTITY;

FUNCTION Objektfolge_konsistent(h0:historisches_Objekt) : BOOLEAN;
...
END_FUNCTION;
```

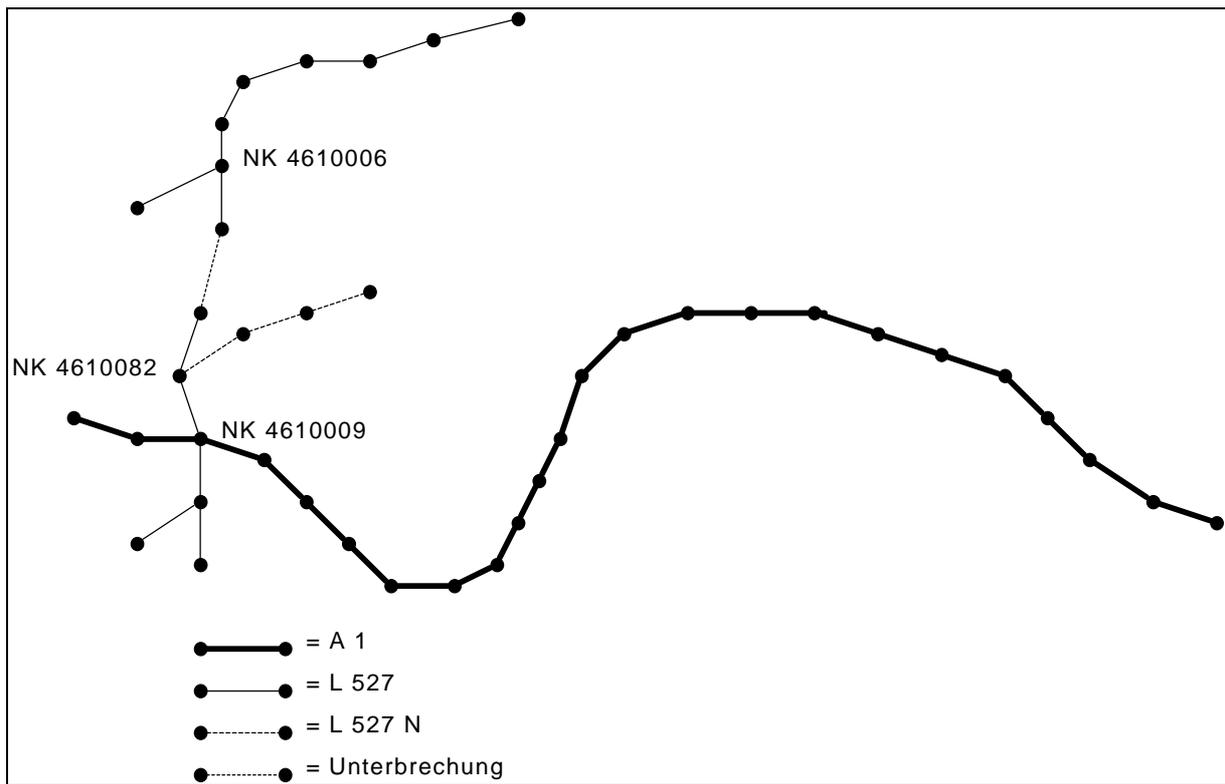
```
ENTITY Ereignis;
--- Attribute :
  laufende_Nummer      : INTEGER;
  Wirksamkeitsdatum   : Datum;
--- Relationen :
  gehoert_zu_Projekt   : OPTIONAL Projekt_Strassenbau;
  in_Bauamt            : SET [1:?] OF Strassenbaudienststelle;
  hat_Dokument_zur_Erlaeuterung : OPTIONAL Dokument;
  hat_identisches_Netzteil : OPTIONAL SET [1:?] OF identisches_Netzteil;
INVERSE
  erzeugt_historisches_Objekt : SET [0:?] OF historisches_Objekt
                                FOR erzeugt_von_Ereignis;
  loescht_historisches_Objekt  : SET [0:?] OF historisches_Objekt
                                FOR geloescht_von_Ereignis;
END_ENTITY;
```

#### 2.1.2 Darstellung einer *Netznotenfolge*

Wir betrachten die (rein schematische und nicht wirklichkeitsgetreue Darstellung der) Umgebung einer Kreuzung zwischen der A1 und der L527:

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht E - Datenaustausch



Im folgenden wird eine Darstellung in der **DA100** gegeben. Zum besseren Verständnis sollen hier die relevanten Einträge am Beispiel des ersten aufgenommenen Netzknotens kurz angegeben werden (für eine ausführlichere Erläuterung siehe ASB92). Die Zahlen unter der Zeile geben ausgewählte Positionen an:

```

100A 0001 105      0590000014709003      12      0101
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
1  4  |  12  |  21  25  29  |  39

```

Spalte	Inhalt
1-3	Datenart (hier: 100)
4-10	Straße
4	Gruppe
6-9	Nummer
10	Buchstabe
11	Widmung
12-19	Verwaltungsbezirk
12-13	Land
14	Regierungsbezirk
15-16	Landkreis bzw. kreisfreie Stadt

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht E - Datenaustausch

17-19	Gemeinde bzw. Stadt
21-23	Straßenbauamt bzw. Autobahnamt Dienststellenschlüssel
25-27	Laufende Nummer
29-35	Netzknoten
39	Verlaufskennzeichen
40	Knotenart

Darstellung der Netzknotenfolge in der DA100:

100A 0001 105	0590000014709003	12	0101
100A 0001 105	0590000024609011	2	0101
100A 0001 105	0590000034610009	2	0101
100A 0001 105	0590000054610021	2	0101
100A 0001 105	0590000064510011	2	0101
100A 0001 105	0590000074511041	2	0101
100A 0001 105	0590000084511033	2	0101
100A 0001 105	0590000094411094	2	0101
100A 0001 105	0590000104411025	2	0101
100A 0001 105	0590000114412044	2	0101
100A 0001 105	0590000124312017	2	0101
100A 0001 105	0590000134312044	2	0101
100A 0001 105	0590000144312014	2	0101
100A 0001 105	0590000154211021	2	0101
100A 0001 105	0590000164011006	2	0101
100A 0001 105	0590000174011079	2	0101
100A 0001 105	0590000183911030	2	0101
100A 0001 105	0590000193812015	2	0101
100A 0001 105	0590000203712041	2	0101
100A 0001 105	0590000213713038	2	0101
100A 0001 105	0543000223713041		0101
100A 0001 105	0543000233613255	2	0101
100A 0001 105	0543000243613027		0101
100A 0001 105	0590000253613026	2	0101
100L 0527 105	0592000014709006	1	0101
100L 0527 105	0592000024709015	1	0101
100L 0527 105	0592000034709013	1	0101
100L 0527 105	0592000044709019	1	0905
100L 0527 105	0592000054709145	1	0905
100L 0527 105	0592000064609015	1	0905
100L 0527 105	0592000074610006	31	0611
100L 0527 105	0592000084610090	41	0611
100L 0527 105	0592000094610006	91	0611
100L 0527 105	0592000104610006	51	0611
100L 0527 105	0592000114610008	61	0101
100L 0527 105	0592000124610082	1	0101
100L 0527 105	0592000134610009	2	0101
100L 0527 105	0592000144610087	31	0101
100L 0527 105	0592000154610019	41	0101
100L 0527 105	0592000164610087	91	0101
100L 0527 105	0592000174510009	21	0101
100L 0527N105	0592000014610083	11	0101
100L 0527N105	0592000024610081	1	0101
100L 0527N105	0592000034610082	21	0101

Als Vergleich dazu geben wir hier die Darstellung in CTE (ISO 10303-21) auf Basis des Entwurfs zum OKSTRA-Schema (siehe 2.1.1). Für den Rahmen dieses Beispiels gehen wir davon aus, daß die gesamten Daten ab dem 1. September 1998 wirksam sind.

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht E - Datenaustausch

```
#1 = Strasse ( '01.09.1998', $, #20, $, $, #11, ( #101, #102, #103, #105, #106, #107, #108,
#109, #110, #111, #112, #113, #114, #115, #116, #117, #118, #119, #120, #121,
#122, #123, #124 ), ( #31 ), $, ( #41 ));
#3 = Strasse ( '01.09.1998', $, #20, $, $, #13, ( #301, #302, #303, #304, #305, #306, #307,
#308, #309, #310, #311, #312, #313, #314, #315, #316, #317 ), ( #33 ), $,
( #41 ));
#4 = Strasse ( '01.09.1998', $, #20, $, $, #14, ( #401, #402, #403 ), ( #34 ), $, ( #41 ));

#11 = Strassenbezeichnung ( #2001, 1, $ );
#13 = Strassenbezeichnung ( #2003, 527, $ );
#14 = Strassenbezeichnung ( #2003, 527, 'N' );

#20 = Ereignis ( ... );

#31 = Abschnitt_oder_Ast ( ... );
#33 = Abschnitt_oder_Ast ( ... );
#34 = Abschnitt_oder_Ast ( ... );

#41 = Bundesland ( ... , 05 );

#101 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3001, ( #601 ), $ );
#102 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #602 ), ( #101 ));
#103 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #603 ), ( #102 ));
#105 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #605 ), ( #103 ));
#106 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #606 ), ( #105 ));
#107 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #607 ), ( #106 ));
#108 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #608 ), ( #107 ));
#109 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #609 ), ( #108 ));
#110 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #610 ), ( #109 ));
#111 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #611 ), ( #110 ));
#112 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #612 ), ( #111 ));
#113 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #613 ), ( #112 ));
#114 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #614 ), ( #113 ));
#115 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #615 ), ( #114 ));
#116 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #616 ), ( #115 ));
#117 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #617 ), ( #116 ));
#118 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #618 ), ( #117 ));
#119 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #619 ), ( #118 ));
#120 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #620 ), ( #119 ));
#121 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #621 ), ( #120 ));
#122 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #622 ), ( #121 ));
#123 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #623 ), ( #122 ));
#124 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #624 ), ( #123 ));
#125 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3002, ( #625 ), ( #124 ));

#301 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3001, ( #701 ), $ );
#302 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #702 ), ( #301 ));
#303 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #703 ), ( #302 ));
#304 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #704 ), ( #303 ));
#305 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #705 ), ( #304 ));
#306 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #706 ), ( #305 ));
#307 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3003, ( #707 ), ( #306 ));
#308 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3004, ( #708 ), ( #307 ));
#309 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3009, ( #707 ), ( #308 ));
#310 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3005, ( #707 ), ( #309 ));
#311 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3006, ( #711 ), ( #310 ));
#312 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #712 ), ( #311 ));
#313 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #603 ), ( #312 ));
#314 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3003, ( #714 ), ( #313 ));
#315 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3004, ( #715 ), ( #314 ));
#316 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3009, ( #714 ), ( #315 ));
#317 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3002, ( #717 ), ( #316 ));

#401 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3001, ( #801 ), $ );
#402 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, $, ( #802 ), ( #401 ));
#403 = Strasse_Netzknoden ( '01.09.1998', $, #20, $, $, #3002, ( #712 ), ( #402 ));

#601 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4709, 003, #4002, $, $, $, $ );
#602 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4609, 011, #4002, $, $, $, $ );
#603 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 009, #4002, $, $, $, $ );
#605 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 021, #4002, $, $, $, $ );
#606 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4510, 011, #4002, $, $, $, $ );
#607 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4511, 041, #4002, $, $, $, $ );
#608 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4511, 033, #4002, $, $, $, $ );
#609 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4411, 094, #4002, $, $, $, $ );
#610 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4411, 025, #4002, $, $, $, $ );
#611 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4412, 044, #4002, $, $, $, $ );
#612 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4312, 017, #4002, $, $, $, $ );
#613 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4312, 044, #4002, $, $, $, $ );
```

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht E - Datenaustausch

---

```
#614 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4312, 014, #4002, $, $, $, $ );
#615 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4211, 021, #4002, $, $, $, $ );
#616 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4011, 006, #4002, $, $, $, $ );
#617 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4011, 079, #4002, $, $, $, $ );
#618 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 3911, 030, #4002, $, $, $, $ );
#619 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 3812, 015, #4002, $, $, $, $ );
#620 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 3712, 041, #4002, $, $, $, $ );
#621 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 3713, 038, #4002, $, $, $, $ );
#622 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 3713, 041, #4000, $, $, $, $ );
#623 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 3613, 255, #4002, $, $, $, $ );
#624 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 3613, 027, #4000, $, $, $, $ );
#625 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 3613, 026, #4000, $, $, $, $ );

#701 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4709, 006, #4000, $, $, $, $ );
#702 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4709, 015, #4001, $, $, $, $ );
#703 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4709, 013, #4001, $, $, $, $ );
#704 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4709, 019, #4001, $, $, $, $ );
#705 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4709, 145, #4001, $, $, $, $ );
#706 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4609, 015, #4001, $, $, $, $ );
#707 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 006, #4001, $, $, $, $ );
#708 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 009, #4001, $, $, $, $ );
#711 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 008, #4001, $, $, $, $ );
#712 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 082, #4001, $, $, $, $ );
#714 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 087, #4001, $, $, $, $ );
#715 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 019, #4001, $, $, $, $ );
#717 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4510, 009, #4001, $, $, $, $ );

#801 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 083, #4001, $, $, $, $ );
#802 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, 4610, 081, #4001, $, $, $, $ );

#2001 = Strassenklasse ( 'A', 'Bundesautobahn' );
#2002 = Strassenklasse ( 'B', 'Bundesstrasse' );
#2003 = Strassenklasse ( 'L', 'Landstrasse' );
#2004 = Strassenklasse ( 'S', 'Staatsstrasse' );
#2005 = Strassenklasse ( 'K', 'Kreisstrasse' );
#2006 = Strassenklasse ( 'G', 'Gemeindestrasse' );

#3001 = Verlaufskennzeichen ( 1, 'Anfang Strasse' );
#3002 = Verlaufskennzeichen ( 2, 'Ende Strasse' );
#3003 = Verlaufskennzeichen ( 3, 'Anfang Seitenarm' );
#3004 = Verlaufskennzeichen ( 4, 'Ende Seitenarm' );
#3005 = Verlaufskennzeichen ( 5, 'Anfang Unterbrechung' );
#3006 = Verlaufskennzeichen ( 6, 'Ende Unterbrechung' );
#3007 = Verlaufskennzeichen ( 7, 'Anfang getr. verl. Fahrbahn' );
#3008 = Verlaufskennzeichen ( 8, 'Ende getr. verl. Fahrbahn' );
#3009 = Verlaufskennzeichen ( 9, 'Wiederholung' );

#4000 = Knotenart ( 0, 'fiktiver Netzknoten' );
#4001 = Knotenart ( 1, 'plangleich (hoehengleich)' );
#4002 = Knotenart ( 2, 'planfrei (hoehenungleich)' );
#4003 = Knotenart ( 3, 'teilplanfrei' );
```

Anmerkung: Durch die Integration der Historisierung im Datenschema des OKSTRA wird die CTE-Darstellung deutlich verkompliziert, da jedes historisierende Objekt fünf weitere Einträge erhält. Auf der anderen Seite ist dadurch zusätzliche Information direkt im CTE-Eintrag eines Objektes enthalten.

### 3 Aspekte zum Datenaustausch zwischen OKSTRA-Implementierungen

Die Wandlung zwischen Daten in Implementierungen des OKSTRA (beispielsweise einer SQL-Datenbank) und Daten in CTE in einem realen Arbeitsumfeld soll hier anhand von Beispielen kurz angerissen werden. Die Grundlage dazu bilden wieder die EXPRESS-Definitionen aus Abschnitt 2.1.1.

Eine einfache Abfrage wäre zum Beispiel, für ein Objekt die zugehörigen Attribute auszugeben, d.h. Eigenschaften, die gemäß dem EXPRESS-Schema direkt in das Objekt eingetragen werden statt ein anderes Objekt zu referenzieren. Für einen *Netzknoten* wären das der *Numerierungsbezirk*, die *Nummer* und der *Knotenname*. Eine Ausgabe in CTE könnte folgendermaßen aussehen:

```
#1 = Netzknoten ( $, $, '01.09.1998', $, #20, $, $, $, 1234, 012,
                'Autobahnkreuz Adorf', $, $, $ );
```

Das '\$'-Zeichen steht dabei als Platzhalter für nicht besetzte Felder. Wie schon in obigem Beispiel ist hier zu beachten, daß zuerst die Attribute von SUPERTYPES in der im Objekt angegebenen Reihenfolge eingetragen werden, und erst dann die Attribute und Relationen des Objekts selber. So stammen die ersten beiden Einträge für den *Netzknoten* von dem SUPERTYPE *Punktobjekt\_Modell* und die fünf folgenden von dem SUPERTYPE *historisches\_Objekt*. Die hinteren sechs Einträge enthalten die Attribute des *Netzknotens*.

Eine komplexere Abfrage wäre, für ein Objekt den Unterbaum, der gemäß dem OKSTRA-Datenschema in diesem Objekt seine Wurzel hat, auszugeben. Für eine *Straße* könnte der Beginn der Ausgabe folgendermaßen aussehen:

```
#1 = Strasse( '01.09.1998', $, #2, $, $, #3, ( #4, #5 ), ( #6 ), $, ( #7 ) );
#2 = Ereignis ( ... );
#3 = Strassenbezeichnung( #8, 1, 'N' );
#4 = Strasse_Netzknoten( '01.09.1998', $, #2, $, $, #9, ( #10 ), $ );
#5 = Strasse_Netzknoten( '01.09.1998', $, #2, $, $, #11, ( #12 ), $ );
#6 = Abschnitt_oder_Ast ( ... );
#7 = Bundesland ( ... , 05 );
#8 = Strassenklasse( 'B', 'Bundesstrasse' );
#9 = Verlaufskennzeichen( 1, 'Anfang Strasse' );
#10 = Netzknoten( $, $, '01.09.1998', $, #2, $, $, 1234, 012, #13, 'Kreuz Adorf',
                #14, #15, $ );
#11 = Verlaufskennzeichen( 2, 'Ende Strasse' );
#12 = Netzknoten( $, $, '01.09.1998', $, #2, $, $, 4321, 210, #16,
                'Dreieck Edorf', #17, #18 );
#13 = Knotenart( 1, 'plangleich(hoehengleich)' );
#14 = Knotenpunktform( 1, 'Einmuendung oder Kreuzung von 2-streifigen Strassen' );
#15 = Knotenpunktsystem( 1, 'Trompete' );
#16 = Knotenart( 2, 'planfrei(hoehenungleich)' );
#17 = Knotenpunktform( 05, 'Kreuzung 2-streifiger Strassen als Versatz' );
#18 = Knotenpunktsystem( 05, 'Kleeblatt' );
. . .
```

Bei dieser Art der Abfrage ist zu beachten, daß aufgrund von Relationen weitere Objekte ausgegeben werden, die ihrerseits Relationen zu anderen Objekten haben können. Es setzt dadurch ein rekursiver Prozeß ein. Um diesen Prozeß kontrollieren zu können, muß das Programm die Möglichkeit bieten, Abfragemuster zu berücksichtigen. Das bedeutet, man kann festlegen, wie weit die Abfrageroutine entlang der unterschiedlichen Relationen laufen soll, bevor sie das rekursive Verfahren abbricht. Im obigen Beispiel einer *Straße* wird bezüglich des *historischen\_Objekts* grundsätzlich mit dem *Ereignis* abgebrochen. Den Zweig entlang der Relation *hat\_Strassenbezeichnung* kann man in diesem Sinne unbeschränkt lassen, da er nach zwei Schritten von selbst abbricht. Der Zweig *hat\_Strassenknoten* wird

durch das Objekt *Netzknoten* begrenzt. Entlang den Relationen *hat\_Abschnitt\_oder\_Ast* und *in\_Verwaltungsbezirk* wird mit den entsprechenden Objekten direkt abgebrochen. Die Relation *hat\_BAB\_Knotennummer* wird nicht berücksichtigt.

Im allgemeinen wird man jedoch nicht alle Objekte einer Klasse ausgeben wollen, d.h. man wird Bedingungen an die Attribute des Objektes selbst als auch an die Attribute derjenigen Objekte, zu denen das Objekt Relationen hat, stellen. Ausgegeben wird dann nur die Teilmenge derjenigen Objekte, die diesen Bedingungen genügen.

Nach Einstellung der erforderlichen Bedingungen würde die Exportfunktion die gewünschten Daten aus einer Implementierung des OKSTRA (etwa einer SQL-Datenbank) herausuchen und in *Clear Text Encoding* darstellen. In umgekehrter Richtung müßte das Programm in der Lage sein, in CTE gegebene Daten so aufzubereiten (etwa in SQL-Statements umzuwandeln), daß ein Pflegeprogramm einer Implementierung des OKSTRA sie verwenden könnte<sup>1</sup>.

Aus dieser Diskussion ergibt sich ein Problem in der aktuellen Formulierung des OKSTRAs als ein in sich geschlossenes konzeptuelles Schema. Diese Geschlossenheit verbietet ein „kappen“ an definierter Stelle – zumindest sofern Pflichtrelationen betroffen sind. Schließlich müssen beim Austausch eines Entities alle Bedingungen des EXPRESS-Schemas erfüllt sein. Im Ergebnis müßte man stets komplette Datenbestände austauschen; häufig wird man sich aber, wie oben erläutert wurde, auf einen Teilbereich beschränken wollen. Der Grund dafür, daß man früher aufhören kann, ist, daß etliche Objekte ein hinreichendes Eigenleben führen und entsprechend durch eigene Namen eindeutig identifizierbar sind (Nullpunkte, Netzknoten, Straßen usw. tragen alle eindeutige Namen). Zur Integration dieser Sollbruchstellen in das EXPRESS-Schema sind derzeit zwei Ansätze vorhanden; zum einen könnte man die "Sollbruchstellen"-Relationen als OPTIONAL auszeichnen, zum anderen könnte man bei diesen Relationen neben der Referenz auf das Partnerentity alternativ die Angabe eines Identifikators für das Entity erlauben (d.h. man würde nicht auf den Netzknoten mit all seinen Eigenschaften verweisen, sondern nur auf einen Identifikator "1234567" über den sich der zugehörige Netzknoten leicht auffinden läßt). Die Entscheidung, wie hier zu verfahren ist, sollte erst nach ersten Erfahrungen mit dem OKSTRA-Datenaustausch erfolgen.

---

<sup>1</sup> Die Frage, ob solche Daten dann bestehende Daten überschreiben oder ergänzen bzw. stets neue Daten darstellen sollen, ist nicht im Rahmen der Standardisierung eines Austauschverfahrens zu klären, sondern eine Implementierungsfrage für OKSTRA-konforme Datenbanken.