



Objektkatalog für das Straßen- und Verkehrswesen Integration symbolischer Verweise in den OKSTRA®

Version: 1.01

Datum: 18.12.2000

Status: in Bearb.

Dateiname: N0015.doc

Pfad: n.a.

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

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



1 Allgemeines

1.1 Inhaltsverzeichnis

1 Allgemeines	2
1.1 Inhaltsverzeichnis	2
1.2 Abkürzungen und Definitionen	2
1.3 Abbildungsverzeichnis	2
1.4 Tabellenverzeichnis.....	2
1.5 Bezüge	2
1.6 Änderungen	3
1.7 Bearbeitungsvermerke.....	3
2 Zweck des Dokuments	4
2.1 Leserkreis	4
2.2 Kernaussagen des Inhalts	4
2.3 Problemstellung.....	4
2.4 Zur fachlichen Frage	5
2.5 Zur technischen Frage.....	5
2.6 Realisierung mittels Vererbung.....	6
2.6.1 Umsetzung in EXPRESS.....	6
2.7 Auswirkungen auf das SQL-Schema des OKSTRA®	8
2.8 Auswirkungen auf die Rückrelation	9
2.8.1 Umsetzung in EXPRESS.....	9
2.9 Alternativen.....	10

1.2 Abkürzungen und Definitionen

OKSTRA® Objektkatalog für das Straßen- und Verkehrswesen

1.3 Abbildungsverzeichnis

keine Abbildungen

1.4 Tabellenverzeichnis

Tabelle 1 - berücksichtigte OKSTRA®-Objekte.....5

1.5 Bezüge

Dokument	Bemerkungen
OKSTRA®-Website	http://www.okstra.de
T0001	Einführung in die NIAM-Technik, zu finden auf der OKSTRA®-



Dokument	Bemerkungen
	Website unter <i>Datenschema</i>

1.6 Änderungen

Name	Datum	Kapitel	Bemerkungen	Bearbeiter
I0007	20.07.2000	alle	Dokument als Diskussionspapier erstellt	Dietmar König
I0007	25.07.2000	n.a.	Auswirkungen auf das SQL-Schema	Dietmar König
I0007	01.08.2000	alle	Überarbeitung zur Abstimmung durch BG10	Dietmar König
N0015	18.09.2000	alle	Überführung in Dokument N0015 zur Veröffentlichung im Internet	Dietmar König
N0015	18.12.2000	2.8	Betrachtung der Rückrelation	Dietmar König

1.7 Bearbeitungsvermerke

keine



2 Zweck des Dokuments

2.1 Leserkreis

Dieses Dokument richtet sich an alle am OKSTRA® interessierten Kreise.

Vorausgesetzt werden Kenntnisse

- des OKSTRA®-Standards in seiner aktuellen Fassung,
- zum OKSTRA® und seinen Regularien (siehe hierzu auch die OKSTRA®-Website unter <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.

2.2 Kernaussagen des Inhalts

Das Dokument stellt einen Vorschlag zur Erweiterung der Möglichkeiten des OKSTRA® dar.

An zentralen Stellen im Datenschema, vor allen Dingen im Straßennetz, werden Möglichkeiten geschaffen statt der Objekte selbst nur deren Kennungen auszutauschen. Durch die Integration symbolischer Verweise, vor allem an zentralen Stellen des Netzes, wird der Austausch OKSTRA®-konformer Daten erheblich erleichtert.

2.3 Problemstellung

Problem: Die Anforderungen an die referentielle Integrität im OKSTRA®-Schema sind sehr hoch. Es existieren viele Pflichtrelationen. Das führt dazu, dass OKSTRA®-Daten weit aus umfangreicher werden können als gewünscht.

Beispiel: Ein Abschnitt oder Ast muss gemäß dem OKSTRA®-Schema seine begrenzenden Nullpunkte mit austauschen. Jeder dieser Nullpunkte wiederum benötigt seinen Netzknoten und mindestens einen Nullpunktsort. Ferner benötigt ein Abschnitt oder Ast noch seine Straße, die (strenggenommen) ihre komplette Netzknotenfolge mit austauschen muss.

Maßnahme: An geeigneten Stellen im Datenschema müssen Möglichkeiten geschaffen werden einen Datensatz abzurunden ohne weitere Objekte hinzuzufügen. Für obiges Beispiel genügt es für den Abschnitt oder Ast zu wissen an welchen Nullpunkten er beginnt und endet bzw. zu welcher Straße er gehört.

Konzept: Die beschriebenen Schnitte können dort eingeführt werden, wo das Partner-Objekt durch eine textliche oder ähnliche Beschreibung eindeutig identifiziert werden kann. Der Datenlieferant teilt darüber dem Datenempfänger mit, welches Partner-Objekt er meint ohne das Objekt selbst mit auszutauschen. Man erhält quasi eine "Steckdose", die man wieder mit dem Partner-Objekt verbinden kann sofern man es besitzt.

In diesem Dokument werden die beiden wesentlichen Fragen beantwortet:

- fachlich: An welchen Stellen im Datenschema sind diese symbolischen Verweise sinnvoll?

- technisch: Wie werden die symbolischen Verweise im Datenschema realisiert?

2.4 Zur fachlichen Frage

Generell können symbolische Verweise anstelle von Objekten verwendet werden, wenn die Objekte einen global eindeutigen konzeptionellen Schlüssel besitzen. Die Minimalanforderung ist ein zwischen Datenlieferant und Datenempfänger eindeutiger Schlüssel. Dazwischen gibt es einen großen Spielraum. Hier im ersten Schritt werden die wesentlichen Objekte, die eindeutige konzeptionelle Schlüssel tragen, durch symbolische Verweise verallgemeinert.

Besonders notwendig (und glücklicherweise gerade dort möglich!) sind symbolische Verweise im Straßennetz und der Administration. Lieferanten von fachlichen Daten haben oft nur unvollständige Daten über das Straßennetz selbst. In diesem Fall können sie durch Verwendung symbolischer Verweise dennoch korrekte OKSTRA®-Daten erzeugen.

Für folgende Objekte werden symbolische Verweise mit Hilfe konzeptioneller Schlüssel eingeführt:

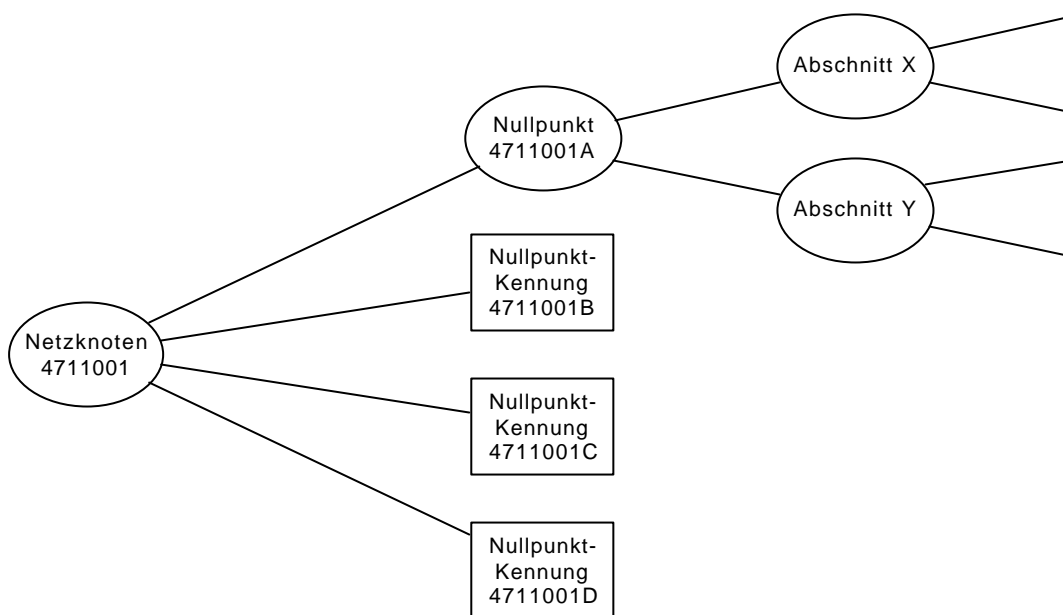
Objekt	konzeptioneller Schlüssel
Netzknoten	7-stelliger Schlüssel, bestehend aus 4-stelliger TK25-Blattnummer und 3-stelliger laufender Nummer
Nullpunkt	8-stelliger Schlüssel, bestehend aus 7-stelligem Netzknoten-Schlüssel und 1-stelligem Zusatzbuchstaben
Abschnitt oder Ast	16-stelliger Schlüssel, gebildet aus 8-stelligem Nullpunkt-Schlüssel am Anfang und 8-stelligem Nullpunkt-Schlüssel am Ende
Straße	Schlüssel aus der Straßenklasse (1 Buchstabe), der Straßenummer (1-3 Ziffern) und evtl. 1 Zusatzbuchstaben, optional mit Länderkennung (2 Zeichen)
Verwaltungsbezirk	konzeptioneller Schlüssel (2-10 Zeichen)
Straßenbaudienststelle	konzeptioneller Schlüssel (2-6 Zeichen)
Bauwerk	7-stellige Bauwerksnummer
Teilbauwerk	9-stelliger Schlüssel, gebildet aus der 7-stelligen Bauwerksnummer und einer 2-stelligen Teilbauwerksnummer
automatische Dauerzählstelle	8-stelliger Schlüssel, gebildet aus der 4-stelligen TK25-Blattnummer und einer 4-stelligen laufenden Nummer
manuelle Zählstelle	8-stelliger Schlüssel, gebildet aus der 4-stelligen TK25-Blattnummer und einer 4-stelligen laufenden Nummer

Tabelle 1 - berücksichtigte OKSTRA®-Objekte

2.5 Zur technischen Frage

Die grundsätzliche Auswirkung dieser Maßnahme ist, dass anstelle eines direkten Verweises auf einen Relationspartner wahlweise auch ein Platzhalter stehen kann.

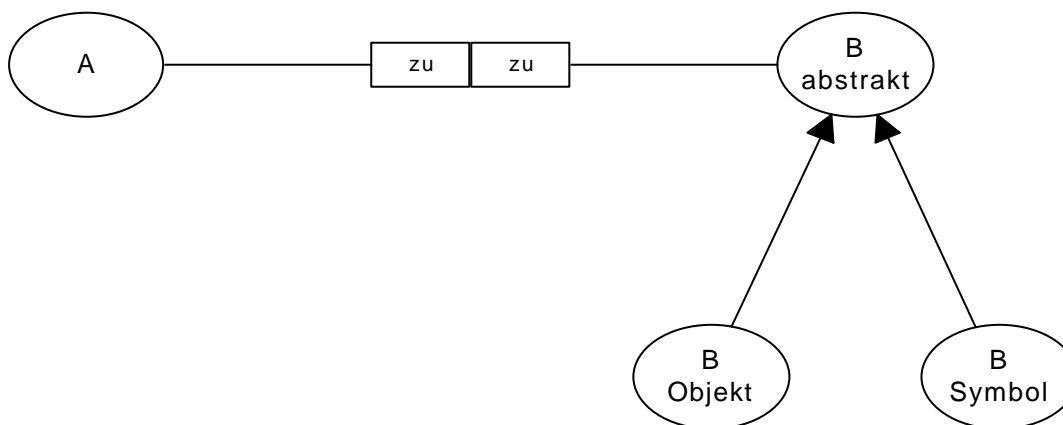
Praktisches Beispiel: Ein Netzknoten beschreibt seine Nullpunkte. Einer der Nullpunkte wird dabei als Nullpunkt-Objekt erfasst, die anderen nur über ihre Kennung. Schematisch sieht das so aus:



Für jedes Partner-Objekt kann individuell entschieden werden, ob das Objekt selbst oder ein symbolischer Verweis aufgenommen werden soll. Im folgenden Kapitel wird die technische Umsetzung dieses Konzepts im OKSTRA®-Schema erläutert.

2.6 Realisierung mittels Vererbung

In der folgenden Skizze besitzt A eine Relation zu B. An dieser Stelle soll im Datenschema eine Sollbruchstelle eingerichtet werden. Für B müssen wir daher die Möglichkeit eines symbolischen Verweises schaffen. Das ursprüngliche Objekt B ist in dieser Konstruktion als B_Objekt eingebettet.



Der Verweis bezieht sich statt auf das ursprüngliche B-Objekt nun auf eine Oberklasse B-abstrakt. Jede Instanz von B_abstrakt ist dann entweder ein B_Objekt oder ein B_Symbol.

2.6.1 Umsetzung in EXPRESS

In EXPRESS stellt sich diese Situation wie folgt dar:



```
ENTITY A;
--- Attribute :
--- Relationen :
    Relation_zu_B          : OPTIONAL SET[1:?] OF B_abstrakt;
END_ENTITY;

ENTITY B_abstrakt
ABSTRACT SUPERTYPE OF (ONEOF(B_Objekt,B_Symbol));
--- Attribute :
--- Relationen :
INVERSE
    Relation_zu_A          : SET[0:?] OF A FOR Relation_zu_B;
END_ENTITY;

ENTITY B_Objekt
SUBTYPE OF (B_abstrakt);
--- Attribute :
    Attribut_von_B         : INTEGER;          -- Beispiel
--- Relationen :
    Relation_zu_X          : X;                -- Beispiel
    Relation_zu_Y          : SET[1:?] OF Y;    -- Beispiel
    Relation_zu_Z          : SET[1:?] OF Z;    -- Beispiel
END_ENTITY;

ENTITY B_Symbol
SUBTYPE OF (B_abstrakt);
--- Attribute :
    Kennung_von_B          : STRING(7);      -- Beispiel
--- Relationen :
END_ENTITY;
```

Klarer wird der Unterschied wenn man einen typischen Ausschnitt aus einem CTE-File betrachtet:

```
#1 = A(#2,#10,#11);

#2 = B_Objekt(2,#3,(#4,#5,#6),(#7,#8,#9));

#3 = X(3,3,3); /* Beispiele für Attribute */
#4 = Y(4,4,4); /* Beispiele für Attribute */
#5 = Y(5,5,5); /* Beispiele für Attribute */
#6 = Y(6,6,6); /* Beispiele für Attribute */
#7 = Z(7,7,7); /* Beispiele für Attribute */
#8 = Z(8,8,8); /* Beispiele für Attribute */
#9 = Z(9,9,9); /* Beispiele für Attribute */

#10 = B_Symbol("CDEFGHI");

#11 = B_Symbol("JKLMNOP");
```

Die Referenz auf das B_Objekt zieht nach sich, dass man auch die Referenzen auf die Partner-Objekte X, Y und Z von B mit aufnehmen muss. Diese könnten weitere Referenzen enthalten usw. Auf diese Weise kann eine Kettenreaktion von Objektreferenzen in Gang gesetzt werden, obwohl man eigentlich nur am Bezug zum Objekt B interessiert ist.

Diesem Umstand trägt das Objekt B_Symbol Rechnung. Es ersetzt den Bezug zum B_Objekt durch eine (gekapselte) Kennung. Dadurch zieht der Bezug zu B keine weiteren Referenzen nach sich.



2.7 Auswirkungen auf das SQL-Schema des OKSTRA®

Im folgenden werden die Auswirkungen der Integration symbolischer Verweise in das OKSTRA®-Schema auf das SQL-Schema des OKSTRA® dargestellt. Unter den verschiedenen Möglichkeiten der Abbildung nach SQL wurde hier diejenige gewählt, die die Idee der symbolischen Verweise semantisch am besten wiedergibt. Es ist eine Standardabbildung für abstrakte Supertypes:

```
CREATE TABLE A
(
  A_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE A__Relation_zu_B
(
  A
    INTEGER NOT NULL REFERENCES A,
  B_abstrakt
    INTEGER NOT NULL REFERENCES B_abstrakt
)

CREATE TABLE B_abstrakt
(
  B_abstrakt_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_B_abstrakt
    INTEGER NOT NULL REFERENCES Art_B_abstrakt,
)

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

INSERT INTO Art_B_abstrakt VALUES(1,"B_Objekt")
INSERT INTO Art_B_abstrakt VALUES(2,"B_Symbol")

CREATE TABLE B_Objekt
(
  B_Objekt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (B_Objekt_ID) REFERENCES B_abstrakt,
  Attribut_von_B
    INTEGER NOT NULL,
  Relation_zu_X
    INTEGER NOT NULL REFERENCES X
)

CREATE TABLE B__Relation_zu_Y
(
  B_Objekt
    INTEGER NOT NULL REFERENCES B_Objekt,
  Y
    INTEGER NOT NULL REFERENCES Y
)
```



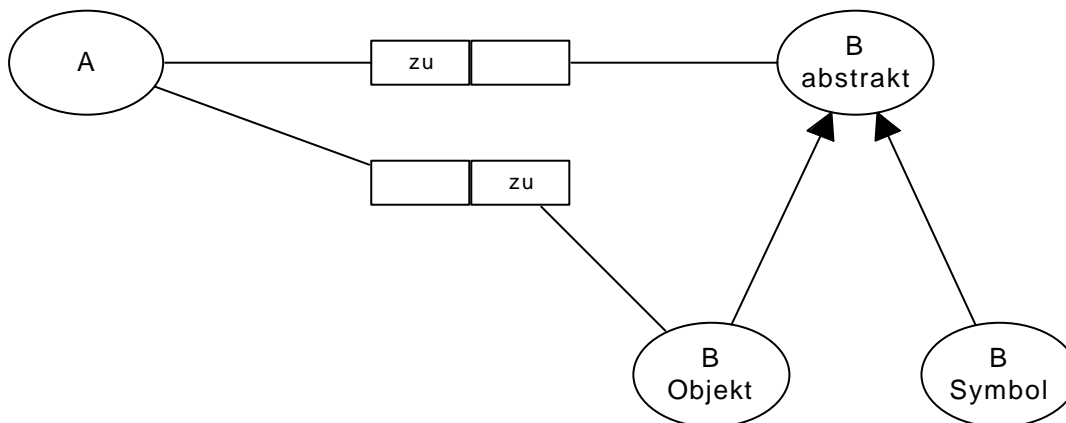

```
CREATE TABLE B_Relation_zu_Z
(
  B_Objekt
    INTEGER NOT NULL REFERENCES B_Objekt,
  Z
    INTEGER NOT NULL REFERENCES Z
)
```

```
CREATE TABLE B_Symbol
(
  B_Symbol_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (B_Symbol_ID) REFERENCES B_abstrakt,
  Kennung_von_B
    VARCHAR(7) NOT NULL
)
```

2.8 Auswirkungen auf die Rückrelation

Obiges Beispiel stellt zunächst isoliert die Änderungen eines Objekts dar, das sich auf ein Objekt mit der Möglichkeit eines symbolischen Verweises bezieht. In obigem Beispiel: A trägt eine Relation zu B, und für B werden symbolische Verweise ermöglicht. Die Rückrichtung der geänderten Relation muss in der Folge gesondert als Relation des ursprünglichen Fachobjekts B zum Objekt A dargestellt werden.

Tatsächlich spaltet sich also die eine beidseitige Relation in zwei einseitige Relationen auf:



2.8.1 Umsetzung in EXPRESS

Tatsächlich stellt sich die Situation in EXPRESS also wie folgt dar:

```
ENTITY A;
--- Attribute :
--- Relationen :
  Relation_zu_B : OPTIONAL SET[1:?] OF B_abstrakt;
END_ENTITY;

ENTITY B_abstrakt
ABSTRACT SUPERTYPE OF (ONEOF(B_Objekt,B_Symbol));
```



```
--- Attribute :  
--- Relationen :  
END_ENTITY;  
  
ENTITY B_Objekt  
SUBTYPE OF (B_abstrakt);  
--- Attribute :  
    Attribut_von_B          : INTEGER;          -- Beispiel  
--- Relationen :  
    Relation_zu_A          : A;  
    Relation_zu_X          : X;                -- Beispiel  
    Relation_zu_Y          : SET[1:?] OF Y;    -- Beispiel  
    Relation_zu_Z          : SET[1:?] OF Z;    -- Beispiel  
END_ENTITY;  
  
ENTITY B_Symbol  
SUBTYPE OF (B_abstrakt);  
--- Attribute :  
    Kennung_von_B          : STRING(7);      -- Beispiel  
--- Relationen :  
END_ENTITY;
```

Die ursprünglich eine Relation (direkte und INVERSE) zwischen A und B ist also in zwei Relationen (ohne eine INVERSE Relation) aufgespalten worden. Die eine Relation geht von A nach B_abstrakt, die andere Relation von B nach A.

2.9 Alternativen

EXPRESS bietet zur Darstellung einer Auswahlmöglichkeit zwischen Typen eine Alternative zur Vererbung: das SELECT-Statement. Bei einem SELECT-Typ hat man für jedes Element die Möglichkeit zwischen den beteiligten Typen einen auszuwählen. Dieses bietet zwei weitere Möglichkeiten zur Umsetzung symbolischer Verweise (in der Terminologie der Objekte A und B wie oben):

- Die Relation von A zu B geht nicht direkt auf B, sondern auf einen Typen dessen Instanzen entweder ein B oder eine String-Kennung sind.
- Die Relation von A zu B geht nicht direkt auf B, sondern wahlweise auf B oder ein Verweis-Objekt, das eine mglw. komplexe Kennung enthält.

Die erste Variante ist von der Darstellung her die kompakteste, bietet jedoch auch die geringste Flexibilität. Lediglich ein einzelner String ist hier als Kennung möglich.

Die zweite Variante bietet technisch dieselben Möglichkeiten wie die oben vorgestellte Realisierung per Vererbung. Der Nachteil ist jedoch - bei beiden Alternativen - die Verwendung eines bisher nicht verwendeten EXPRESS-Konstrukts (SELECT).

Ein weiterer, wesentlicher Nachteil gegenüber der Realisierung mittels Vererbung ist, dass die Möglichkeit eines symbolischen Verweises nicht am Objekt selbst sichtbar wird, sondern nur auf der anderen Seite der Relation, der Seite der referenzierenden Objekte.

Die Betreuungsgruppe der OKSTRA® Pflege hat sich daher nach eingehender Diskussion für die Realisierung mittels Vererbung entschieden, da sie die transparenteste Lösung des Problems darstellt und nur bekannte und vertraute Sprachmittel verwendet.