



Objektkatalog für das Straßen- und Verkehrswesen

Vorschlag zur Einführung eines Metamodells

Version: 1.1
Datum: 19.01.2011
Status: akzeptiert
Dateiname: N0092.doc
Verantwortlich: J. Hettwer

OKSTRA-Pflegestelle

interactive instruments GmbH
Trierer Straße 70-72
53115 Bonn

<http://www.okstra.de/>

Herr Bernd Weidner
Tel. 0228 91410 74
Fax 0228 91410 90
Email weidner@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 Allgemeines.....	2
0.1 Inhaltsverzeichnis.....	2
1 Zweck des Dokuments	3
1.1 Leserkreis	3
1.2 Kernaussagen des Inhalts	3
2 Motivation	4
3 Vorschlag	6
4 Regeln für das OKSTRA®-Metamodell.....	7

1 Zweck des Dokuments

1.1 Leserkreis

Das Dokument richtet sich an alle am OKSTRA® interessierten Personen und Institutionen.

Vorausgesetzt werden Kenntnisse

- der grundlegenden OKSTRA®-Standards, speziell EXPRESS und OKSTRA®-XML, sowie
- zum OKSTRA® und seinen Regularien (siehe auch <http://www.okstra.de/>).

1.2 Kernaussagen des Inhalts

Es werden eine Reihe von Regeln vorgeschlagen – ein Metamodell -, die bei der konzeptionellen OKSTRA®-Modellierung in EXPRESS zu beachten sind. Abweichungen der derzeitigen OKSTRA®-Modellierung von diesem Metamodell werden im Rahmen der nächsten OKSTRA®-Versionierung beseitigt. Kern des Metamodells ist eine Zuordnung der in den EXPRESS-Schemata auftretenden ENTITIES zu bestimmten Kategorien, für die Regeln hinsichtlich der möglichen Beziehungen zu anderen ENTITIES (Vererbung, Relationen) und der möglichen Attribute aufgestellt werden.

Durch die Einführung eines Metamodells für den OKSTRA® sollen zwei Ziele erreicht werden:

1. Gewährleistung eines einheitlichen strukturellen Aufbaus des OKSTRA®.
2. Vermeidung/Beseitigung von Inkonsistenzen zwischen der konzeptionellen EXPRESS-Modellierung und den daraus abgeleiteten Schemata (insbesondere zwischen der EXPRESS-Modellierung und dem OKSTRA®-XML).

2 Motivation

Der OKSTRA®-Standard besteht aus verschiedenen Komponenten, die sich hinsichtlich ihres Aufgabenbereiches und teilweise auch hinsichtlich der enthaltenen modellbezogenen Informationen voneinander unterscheiden. Da ist als erstes das EXPRESS-Modell zu nennen, eine konzeptionelle Modellierung, die als Referenzmodell für den OKSTRA® fungiert. Dann gibt es die NIAM-Diagramme, die als Hilfsmittel zur Modellierung dienen und einen eher dokumentarischen Charakter besitzen (dies deshalb, weil in Zweifelsfällen die EXPRESS-Modellierung maßgebend ist). Schließlich gibt es noch verschiedene aus der EXPRESS-Modellierung abgeleitete Schemata bzw. Datenformate; zu dieser Gruppe gehören OKSTRA®-CTE, -SQL und -XML.

Während das CTE-Format eine generische Abbildung von EXPRESS darstellt (d.h. hier werden die Strukturen der EXPRESS-Sprache selbst in ein Format überführt, völlig unabhängig vom fachlichen Inhalt der Modellierung), gilt dies für die übrigen Ableitungen nicht: Hier spielen teilweise inhaltliche Aspekte eine Rolle. Beim OKSTRA®-XML werden z.B. Straßennetzbezüge und Historisierung über Type-Extension vererbt; alle übrigen Vererbungsbeziehungen werden durch Einbettung der ererbten Eigenschaften in den Subtypen abgebildet. Teilweise sind auch formale Gesichtspunkte, die in EXPRESS nicht ausgedrückt werden können, für die aus dem EXPRESS-Schema generierten Abbildungen von Bedeutung: Schlüsseltabellen werden z.B. im OKSTRA®-XML anders dargestellt als die sonstigen Objektarten; die EXPRESS-Sprache unterscheidet aber nicht zwischen Schlüsseltabellen-ENTITIES und sonstigen ENTITIES. Außerdem haben Schlüsseltabellen im OKSTRA®-XML stets nur die beiden Attribute „Kennung“ und „Langtext“, unabhängig von Zahl, Bezeichnung und Datentyp der Attribute im EXPRESS-Modell.


Da aus dem EXPRESS-Modell des OKSTRA® nicht-generische Ableitungen vorgenommen werden, die auf bestimmten Annahmen hinsichtlich des Inhaltes des Modells oder zusätzlichen formalen Festlegungen basieren, muss die OKSTRA®-EXPRESS-Modellierung diese Annahmen und Festlegungen berücksichtigen. Tut sie das nicht, treten Inkonsistenzen zwischen dem EXPRESS-Modell und den Ableitungen auf. Dies kann z.B. dazu führen, dass bestimmte Inhalte des EXPRESS-Modells im – abgeleiteten – OKSTRA®-XML-Format nicht transportiert werden können. Beispielsweise besitzt die Schlüsseltabelle *Kronenschaden* aus dem Schema Ökologie derzeit im EXPRESS-Modell neben den Attributen „Kennung“ und „Langtext“ ein weiteres Attribut namens „Schadenseinstufung“, hinter dem sich eine weitere Schlüsseltabelle verbirgt. Diese Schadenseinstufung kann derzeit im OKSTRA®-XML nicht transportiert werden, da dort nur die Attribute „Kennung“ und „Langtext“ abgebildet werden (siehe oben).

Zur Vermeidung von Inkonsistenzen zwischen dem EXPRESS-Modell und den daraus gewonnenen Ableitungen bestehen prinzipiell die folgenden Möglichkeiten:

1. (Ständige) Weiterentwicklung der Abbildungsregeln für die aus dem EXPRESS-Modell abgeleiteten Schemata; ggf. Schaffung von Sonderlösungen (z.B. könnte die Schlüsseltabelle *Kronenschaden* anders abgebildet werden als die übrigen Schlüsseltabellen).
2. Berücksichtigung der für die Abbildungsregeln relevanten thematischen und formalen Festlegungen bereits im EXPRESS-Modell.

Der erste Weg besitzt den Vorteil, dass die EXPRESS-Modellierung praktisch ohne Berücksichtigung der Ableitungen vorgenommen werden kann. Allerdings müssen dadurch erzeugte Inkonsistenzen je nach Bedarf durch eine Weiterentwicklung der Abbildungsregeln für die abgeleiteten Schemata vermieden werden. Diese – ggf. häufiger auftretenden – Weiterentwicklungen können zu größeren strukturellen Unterschieden zwischen den Ableitungen verschiedener OKSTRA®-Versionen und damit zu erhöhten Umstellungsaufwänden führen. Außerdem führen Sonderlösungen für einzelne Probleme generell zu einer schlechteren Handhabbarkeit des OKSTRA® insgesamt.

Sinnvoller erscheint die zweite Variante: Mit ihr kann eine größere strukturelle Einheitlichkeit des OKSTRA® sowohl innerhalb des EXPRESS-Schemas als auch beim Übergang von EXPRESS zu den

	Objektkatalog für das Straßen- und Verkehrswesen Vorschlag zur Einführung eines Metamodells	Seite: 5 von 11 Name: N0092 Stand: 19.01.2011
--	--	--

Ableitungen erreicht werden. Damit wird bereits im Vorfeld verhindert, dass Inkonsistenzen zwischen dem EXPRESS-Modell und den abgeleiteten Schemata auftreten können. Eine ständige Weiterentwicklung der Abbildungsregeln bzw. die Schaffung von Sonderlösungen ist nicht nötig. Der Preis dafür ist, dass bei der EXPRESS-Modellierung bestimmte Regeln beachtet werden müssen, die eine unproblematische Anwendung der Abbildungsregeln für die aus dem EXPRESS-Schema zu generierenden Ableitungen ermöglichen.

3 Vorschlag

Ausgehend von den im Kapitel „Motivation“ beschriebenen Überlegungen wird folgendes vorgeschlagen:

1. Für den OKSTRA® wird das im folgenden Kapitel „Regeln für das OKSTRA®-Metamodell“ aufgeführte Metamodell eingeführt. Dabei handelt es sich um ein Regelwerk, in dem festgelegt ist, was bei der Erstellung des OKSTRA®-Referenzmodells in EXPRESS erlaubt ist und was nicht.
2. Abweichungen des bestehenden OKSTRA®-EXPRESS-Modells von diesem Metamodell werden im Rahmen der nächsten Versionierung beseitigt.

Als Anhang ist diesem Vorschlag eine HTML-Differenzdarstellung beigelegt, aus der ersichtlich ist, welche Änderungen sich durch die Einführung des Metamodells an EXPRESS-Schema der OKSTRA®-Version 1.011 ergeben würden. Teile, die entfallen würden, sind dabei blau und durchgestrichen. Neu hinzugekommene Teile sind gelb hinterlegt.

4 Regeln für das OKSTRA[®]-Metamodell

§ 1 ENTITY-Klassifikation

(1) Jedes ENTITY des OKSTRA-EXPRESS-Schemas wird einer der folgenden Kategorien zugeordnet:

1. vollwertige Objektart, instanzierbar
2. vollwertige Objektart, abstrakt
3. konzeptionelle Objektart
4. Schlüsseltabelle
5. Symbol für abstrakten Verweis
6. Supertyp für abstrakten Verweis
7. Supertyp für die Historisierung
8. Supertyp zur Kennzeichnung des ENTITY-Typs

(2) Eine vollwertige instanzierbare Objektart ist dadurch gekennzeichnet, dass sie eine eigene Objektidentität besitzt. Die Existenz einer Instanz einer vollwertigen instanzierbaren Objektart ist nicht an die Existenz einer anderen Instanz gebunden; sie kann somit prinzipiell unabhängig von anderen Instanzen existieren.

(3) Vollwertige abstrakte Objektarten sind nicht instanzierbare Supertypen vollwertiger instanzierbarer Objektarten. Sie dienen in der Modellierung dazu, zusammengehörige Eigenschaften zu bündeln und geschlossen weiterzuvererben.

(4) Konzeptionelle Objektarten besitzen keine eigene Objektidentität, sondern stellen lediglich attributive Informationen für andere Objektarten bereit. Eine konzeptionelle Objektart verhält sich in vielerlei Hinsicht wie ein elementarer Datentyp; insbesondere ist die Existenz einer Instanz einer konzeptionellen Objektart unmittelbar oder mittelbar an die Existenz einer Instanz einer vollwertigen Objektart gebunden.

(5) Schlüsseltabellen sind Objektarten zur Repräsentierung von Wertekatalogen. Ihre möglichen Attributwerte sind normalerweise vollständig vorgegeben; es besteht aber auch die Möglichkeit, keine Werte anzugeben. Im ersten Fall sind nur die angegebenen Werte zulässig, im zweiten beliebige Werte (da der OKSTRA[®] die möglichen Werte nicht festlegt).

(6) Ein Symbol für einen abstrakten Verweis ist eine Hilfsobjektart, die für einen abstrakten Verweis auf eine vollwertige instanzierbare oder abstrakte Objektart benötigt wird. Ein solches Symbol kann eine eindeutige Kennung des referenzierten Objektes tragen.

(7) Ein Supertyp für einen abstrakten Verweis ist eine Hilfsobjektart, die für einen abstrakten Verweis auf eine vollwertige instanzierbare oder abstrakte Objektart benötigt wird. Durch Referenzierung eines solchen Supertyps kann sowohl eine vollwertige Objektart als auch das zugehörige Symbol referenziert werden.

(8) Als Supertyp für die Historisierung existiert die Objektart *historisches_Objekt*. Von diesem Supertyp erben die historisierbaren vollwertigen Objektarten. Die Objektart *historisches_Objekt* ist ein Spezialfall einer vollwertigen abstrakten Objektart.

(9) Zur Kennzeichnung von ENTITY-Kategorien existieren die folgenden Supertypen:

1. *OKSTRA_Objekt*
2. *OKSTRA_konzept_Objekt*
3. *OKSTRA_Schlüsseltabelle*

Von diesen Supertypen erben direkt oder mittelbar alle ENTITIES der betreffenden Kategorie.

§ 2 Kennzeichnung des ENTITY-Typs im EXPRESS-Modell

- (1) Eine vollwertige instanzierbare Objektart erbt direkt oder mittelbar von der Objektart *OKSTRA_Objekt* und ist nicht als ABSTRACT SUPERTYPE gekennzeichnet.
- (2) Eine vollwertige abstrakte Objektart erbt direkt oder mittelbar von der Objektart *OKSTRA_Objekt* und ist als ABSTRACT SUPERTYPE gekennzeichnet.
- (3) Eine konzeptionelle Objektart erbt direkt oder mittelbar von der Objektart *OKSTRA_konzept_Objekt*.
- (4) Eine Schlüsseltabelle erbt direkt von der Objektart *OKSTRA_Schlüsseltabelle*.
- (5) Der Name eines Symbols für einen abstrakten Verweis setzt sich zusammen aus dem Namen der vollwertigen Objektart, auf die verwiesen werden soll, und dem Suffix *_Symbol*. Dieses Suffix darf nur für Symbole für abstrakte Verweise verwendet werden.
- (6) Der Name eines Supertypen für einen abstrakten Verweis setzt sich zusammen aus dem Namen der vollwertigen Objektart, auf die verwiesen werden soll, und dem Suffix *_abstrakt*. Dieses Suffix darf nur für Supertypen für abstrakte Verweise verwendet werden.
- (7) Die vier Supertypen *historisches_Objekt*, *OKSTRA_Objekt*, *OKSTRA_konzept_Objekt* und *OKSTRA_Schlüsseltabelle* werden über ihre Objektartennamen identifiziert.

§ 3 Vererbung

- (1) Eine vollwertige instanzierbare Objektart kann von folgenden ENTITY-Arten erben:
 1. *OKSTRA_Objekt*
 2. vollwertige Objektart, instanzierbar
 3. vollwertige Objektart, abstrakt
 4. Supertyp für abstrakten Verweis
 5. *historisches_Objekt*
- (2) Eine vollwertige abstrakte Objektart kann von folgenden ENTITY-Arten erben:
 1. *OKSTRA_Objekt*
 2. vollwertige Objektart, abstrakt
 3. Supertyp für abstrakten Verweis
 4. *historisches_Objekt*
- (3) Eine konzeptionelle Objektart kann von folgenden ENTITY-Arten erben:
 1. *OKSTRA_konzept_Objekt*
 2. konzeptionelle Objektart
- (4) Eine Schlüsseltabelle muss von der Objektart *OKSTRA_Schlüsseltabelle* erben; sie kann von keiner anderen Objektart erben.
- (5) Ein Symbol für einen abstrakten Verweis muss von dem zugehörigen Supertyp für den abstrakten Verweis erben; es kann von keiner weiteren Objektart erben.
- (6) Ein Supertyp für einen abstrakten Verweis kann von keiner anderen Objektart erben. Er muss Supertyp für die abstrakt zu referenzierende vollwertige Objektart und das zugehörige Symbol für den abstrakten Verweis sein.

(7) Die Objektart *historisches_Objekt* erbt nur von der Objektart *OKSTRA_Objekt*.

(8) Ein Supertyp zur Kennzeichnung einer ENTITY-Kategorie kann von keiner anderen Objektart erben.

§ 4 Relationen, allgemein

(1) Relationen zwischen Objektarten werden grundsätzlich beidseitig angelegt. Sofern Ausnahmen von dieser Regel bestehen, sind sie in § 5 und § 6 explizit aufgeführt.

§ 5 Relationen vollwertiger Objektarten

(1) Eine vollwertige (instanzierbare oder abstrakte) Objektart kann Relationen zu folgenden Objektarten besitzen:

1. vollwertige (instanzierbare oder abstrakte) Objektarten
2. konzeptionelle Objektarten
3. Schlüsseltabellen
4. Supertypen für abstrakte Verweise

(2) Hat eine vollwertige (instanzierbare oder abstrakte) Objektart eine Relation zu einer konzeptionellen Objektart, dient das konzeptionelle Objekt als attributive Erweiterung der vollwertigen Objektart. In diesem Fall gibt es keine Rückrelation von der konzeptionellen Objektart zur vollwertigen Objektart.

(3) Hat eine vollwertige (instanzierbare oder abstrakte) Objektart eine Relation zu einer Schlüsseltabelle, dient die Schlüsseltabelle als attributive Erweiterung der vollwertigen Objektart. In diesem Fall gibt es keine Rückrelation von der Schlüsseltabelle zur vollwertigen Objektart.

(4) Hat eine vollwertige (instanzierbare oder abstrakte) Objektart eine Relation zu einem Supertypen für einen abstrakten Verweis, gibt es keine direkte Rückrelation. Als Ersatz muss jedoch eine weitere einseitige Relation bestehen. Dafür existieren die folgenden Möglichkeiten:

1. Der vollwertige Subtyp des Supertypen für den abstrakten Verweis verweist einseitig auf die vollwertige Objektart zurück (Relation mit einseitig möglichem abstrakten Verweis).
2. Falls die vollwertige Objektart ihrerseits ebenfalls einen Supertypen für einen abstrakten Verweis besitzt, kann der vollwertige Subtyp des referenzierten Supertypen seinerseits einseitig auf diesen zweiten Supertypen verweisen (Relation mit zweiseitig möglichem abstrakten Verweis).

§ 6 Relationen sonstiger Objektarten


(1) Eine konzeptionelle Objektart kann Relationen zu folgenden Objektarten besitzen:

1. vollwertige (instanzierbare oder abstrakte) Objektarten
2. konzeptionelle Objektarten
3. Schlüsseltabellen
4. Supertypen für abstrakte Verweise

Eine solche Relation hat stets attributiven Charakter und ist immer einseitig; eine Rückrelation ist nicht möglich, da die konzeptionelle Objektart keine eigenständige Objektidentität besitzt.

(2) Schlüsseltabellen, Symbole und Supertypen für abstrakte Verweise können keine Relationen besitzen.

(3) Die Objektart *historisches_Objekt* besitzt Relationen zur Vorgänger- und Nachfolgerversion (d.h. zu anderen *historischen_Objekten*) sowie zwei Relationen zur Objektart *Ereignis*, mit denen das erzeugende und das löschende Ereignis zur Version des historisierbaren Objektes angeben

	Objektkatalog für das Straßen- und Verkehrswesen Vorschlag zur Einführung eines Metamodells	Seite: 10 von 11 Name: N0092 Stand: 19.01.2011
--	--	---

werden können. Die Objektart *Ereignis* ist eine vollwertige instanzierbare Objektart, die nicht von der Objektart *historisches_Objekt* erbt.

(4) Die Objektarten *OKSTRA_konzept_Objekt* und *OKSTRA_Schluesstabelle* besitzen keine Relationen.

(5) Die Objektart *OKSTRA_Objekt* verhält sich im Hinblick auf Relationen wie eine vollwertige Objektart.

§ 7 Attribute

(1) Grundsätzlich können alle Objektarten Attribute unter Verwendung der elementaren Datentypen oder daraus abgeleiteter Typen besitzen.

(2) Jede Schlüsseltabelle besitzt ein STRING-Attribut namens „Kennung“ (ohne Längenangabe) und ein oder mehrere STRING- oder INTEGER-Attribute. Grundsätzlich sind die möglichen Werte der Attribute vollständig vorgegeben, wobei sich die vergebenen Kennungen voneinander unterscheiden müssen. Jede Kennung ist mit je einem Eintrag jedes weiteren Attributes assoziiert, d.h. in einer Instanz mit einer bestimmten Kennung müssen die dazugehörigen Werte der weiteren Attribute eingetragen sein. Falls keine Werte der Attribute vorgegeben sind, sind aus Sicht des OKSTRA® beliebige Attributwerte erlaubt.

(3) Ein Symbol für einen abstrakten Verweis besitzt lediglich ein einziges Attribut namens „Kennung“, in dem der eindeutige Schlüssel des zu referenzierenden Objektes gespeichert wird.

(4) Ein Supertyp für einen abstrakten Verweis kann keine Attribute besitzen.

(5) Die Objektart *historisches_Objekt* besitzt zwei Attribute „gueltig_von“ und „gueltig_bis“, mit denen das Gültigkeitsintervall der jeweiligen Objektversion angegeben werden kann.

(6) Vollwertige Objektarten besitzen eine eigenständige Objektidentität. Sie können daher zur Unterscheidung und zur (Re-)identifikation mit einem eindeutigen Identifikator versehen werden. Dieser Identifikator wird im Attribut „OKSTRA_ID“ der Objektart *OKSTRA_Objekt* gespeichert.


(7) Die Objektarten *OKSTRA_konzept_Objekt* und *OKSTRA_Schluesstabelle* besitzen keine Attribute.

§ 8 Historisierung

(1) Von der Instanz einer historisierbaren (vollwertigen instanzierbaren) Objektart können mehrere Versionen existieren. Diese Versionen müssen unterschiedliche, aufeinander folgende Gültigkeitsintervalle besitzen. Zwischen der Entstehung und dem Untergang der Instanz ist somit zu einem Zeitpunkt genau eine Version gültig. Die einzelnen Versionen sind darüber hinaus über Vorgänger- und Nachfolgerrelationen miteinander verknüpft.

(2) Wenn eine OKSTRA_ID für eine Instanz einer historisierbaren Objektart vergeben wird, ist sie für alle Versionen der Instanz gültig. Mit der OKSTRA_ID allein kann somit nur die Menge der bestehenden Versionen identifiziert werden, nicht aber eine einzelne Version. Die Identifikation einer einzelnen Version erfolgt über eine Kombination aus der OKSTRA_ID und dem jeweiligen Gültigkeitsintervall.

(3) Relationen zu historisierbaren Objektarten sind stets multipel, da sie sich grundsätzlich auf alle Versionen der zu referenzierenden Instanz(en) beziehen. Welche Version oder Versionen zu einem

	Objektkatalog für das Straßen- und Verkehrswesen Vorschlag zur Einführung eines Metamodells	Seite: 11 von 11 Name: N0092 Stand: 19.01.2011
--	--	---

bestimmten Zeitpunkt als Relationspartner relevant sind, muss anhand der Gültigkeitsintervalle geklärt werden. Dies gilt auch für abstrakte Verweise¹.

(5) Bei einer Relation zwischen zwei historisierbaren Objektarten ist es abweichend von (3) auch ausreichend, wenn nur diejenigen Versionen untereinander verknüpft werden, deren Gültigkeitsintervalle sich überlappen. Damit kann die Bestimmung der zu einem bestimmten Zeitpunkt relevanten Relationspartner effizienter durchgeführt werden. Diese Möglichkeit besteht nicht bei abstrakten Verweisen und bei Relationen zwischen historisierbaren und nicht historisierbaren Objektarten.

(6) Instanzen konzeptioneller Objektarten, die von der Version einer historisierbaren Objektart referenziert werden, sind streng an diese gekoppelt. Das Gültigkeitsintervall der Version gilt gleichzeitig für alle von ihr referenzierten Instanzen konzeptioneller Objektarten. Wird eine neue Version der historisierbaren Objektart erzeugt, müssen für diese auch neue Instanzen der referenzierten konzeptionellen Objektarten erzeugt werden.

¹ Zum einen ist es über einen abstrakten Verweis auch möglich, die kompletten Objektinstanzen zu referenzieren; für diesen Fall ist die Multiplizität der Relation nötig, um alle Versionen der historisierbaren Instanz zu referenzieren. Zum zweiten können, sofern die Referenzierung über die Angabe eines konzeptionellen Schlüssels erfolgt, damit zwar prinzipiell mehrere Versionen einer historisierbaren Instanz referenziert werden. Dies funktioniert aber dann nicht mehr, wenn sich ein Schlüssel ändert, ohne dass dadurch die Objektidentität beeinträchtigt wird. Auch hier ist daher die Multiplizität nötig.