

Übergreifende Konzepte			D000.doc
Datum	Dok.	Oks.	Beschreibung der Änderungen
27.11.2008	1.013	1.013	formale Versionierung
09.10.2007	1.012	1.012	Einführung eines Metamodells gemäß N0092 Vorschlag zur Entfernung der Schemata „Ingenieurbauwerke“, „Projektressourcen“ und „Ausstattung“ gemäß N0093
02.08.2006	1.011	1.011	Dokument „Übergreifende Konzepte“ erstellt

Inhalt

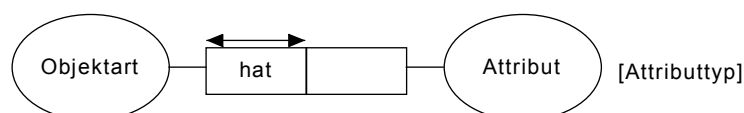
Zweck des Dokumentes	1
Darstellung von Objektarten und Attributen in NIAM-Diagrammen	1
Zentraler Supertype OKSTRA_Objekt, OKSTRA_IDs	2
Verwaltung von IDs aus anderen Fachsystemen.....	2
Abstrakte Verweise	2
Historisierung	4
Geometrie.....	6
Straßennetzbezug von Objekten	6
Grundlegende Typdefinitionen	6

Zweck des Dokumentes

Dieses Dokument beschreibt diejenigen Konzepte des OKSTRA®, die schemaübergreifend sind und deshalb in den NIAM-Dokumenten zu den einzelnen Schemata nicht dargestellt werden. Es ist ein wesentlicher Teil der NIAM-Dokumentation zum OKSTRA®.

Darstellung von Objektarten und Attributen in NIAM-Diagrammen

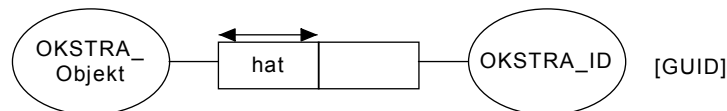
NIAM unterscheidet nicht zwischen Objektarten und Attributen; sie werden daher in gleicher Weise (d.h. durch beschriftete Ellipsen) dargestellt. Die Darstellung der Beziehung zwischen einer Objektart und ihrem Attribut erfolgt analog zur Darstellung einer Relation zwischen zwei Objektarten. Die Rollenbezeichnung auf der Seite der Objektart lautet dabei typischerweise „hat“. Da ein Attribut keine eigenständige Existenz besitzt, bleibt die Rollenbezeichnung der Relation auf seiner Seite leer. Teilweise wird der Datentyp des Attributes in eckigen Klammern neben der Ellipse angegeben:



Attribute werden in den NIAM-Diagrammen zum OKSTRA® häufig nicht dargestellt, um die Darstellung nicht zu überfrachten.

Zentraler Supertype OKSTRA_Objekt, OKSTRA_IDs

Grundsätzlich erben alle Objektarten des OKSTRA® direkt oder mittelbar vom abstrakten Supertyp *OKSTRA_Objekt*, der im Schema „Allgemeine Objekte“ definiert ist. Ausgenommen davon sind nur Schlüsseltabellen und konzeptionelle Objektarten, die ihrer Natur nach einen eher attributiven Charakter besitzen und daher nicht als vollwertige und eigenständige Objektarten betrachtet werden können. Objektarten, die vom *OKSTRA_Objekt* erben, erhalten die Möglichkeit, optional eine weltweit eindeutige OKSTRA_ID zu tragen. Als OKSTRA_IDs sind *Globally Unique Identifiers (GUIDs)* mit einer Länge von 128 Bit (entsprechend 32 Zeichen) zu verwenden.

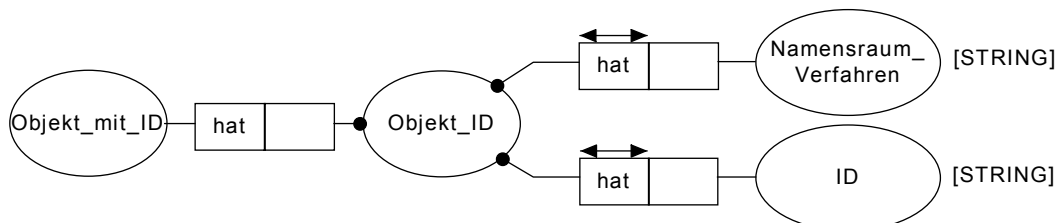


Innerhalb einer *GUID* sind folgende Zeichen zulässig: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. *GUIDs* haben die Eigenschaft, dass sie mittels geeigneter Algorithmen dezentral, aber trotzdem global eindeutig erzeugt werden können.

Die Vererbungsbeziehungen zum *OKSTRA_Objekt* sind in den NIAM-Diagrammen zum OKSTRA® nicht dargestellt, weil dort primär die fachlichen Zusammenhänge beschrieben werden und eine Darstellung der Beziehungen zum *OKSTRA_Objekt* die Diagramme überfrachten würde.

Verwaltung von IDs aus anderen Fachsystemen

Einige OKSTRA®-Objektarten erben vom abstrakten Supertyp *Objekt_mit_ID*, der im Schema „Allgemeine Objekte“ definiert ist. Damit sind sie in der Lage, beliebig viele, aus anderen Fachsystemen stammende Identifikatoren zu verwalten. Ein solcher Identifikator, der durch die Objektart *Objekt_ID* abgebildet wird, besitzt das STRING-Attribut „ID“ zur Aufnahme des eigentlichen Identifikators und das STRING-Attribut „Namensraum_Verfahren“, in dem das erzeugende Fachsystem/Verfahren eingetragen wird.

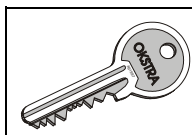


Über den Aufbau und die Länge eines aus einem anderen System stammenden Identifikators trifft der OKSTRA® keine Annahmen. Deshalb ist das Attribut „ID“ der *Objekt_ID* als einfaches STRING-Attribut ohne Längenangabe modelliert.

Abstrakte Verweise

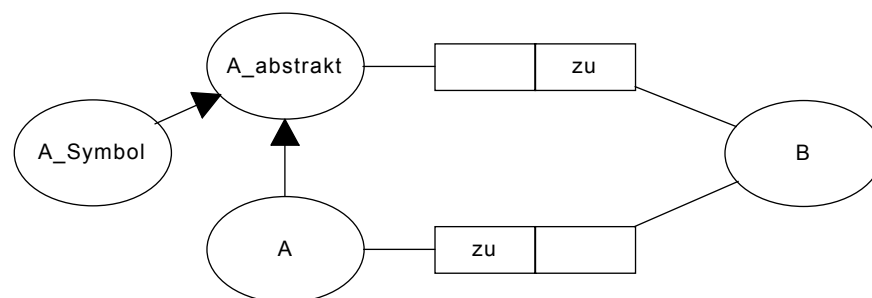
Um anzugeben, dass eine Instanz einer Objektart *A* eine Relation zu einer Instanz einer Objektart *B* besitzt, muss bei Verwendung einer normalen OKSTRA®-Relation die Instanz der Objektart *B* vollständig mit ausgetauscht werden:



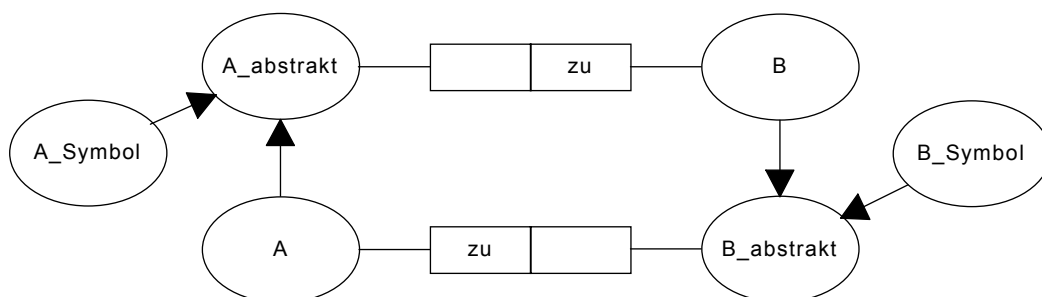


Für manche Anwendungen genügt es allerdings zu wissen, auf welche Instanz sich eine Relation bezieht, ohne dass die Instanz selbst mit ausgetauscht werden muss. Hier kann ein abstrakter Verweis eingesetzt werden. Er erlaubt es, anstelle eines Relationspartners einen konzeptionellen Schlüssel (d.h. einen STRING) anzugeben, der die Identität des Relationspartners eindeutig festlegt.

Um abstrakte Verweise auf eine Objektart *A* zu ermöglichen, wird im OKSTRA® für diese Objektart ein abstrakter Supertyp namens *A_abstrakt* eingeführt, von dem sowohl die Objektart *A* als auch eine Objektart *A_Symbol* erben. Die Objektart *A_Symbol* erhält lediglich ein STRING-Attribut zur Aufnahme des konzeptionellen Schlüssels. Soll von einer Objektart *B* aus ein abstrakter Verweis auf die Objektart *A* möglich sein, wird die dafür vorgesehene Relation nicht auf die Objektart *A*, sondern auf die Objektart *A_abstrakt* gerichtet. Damit kann von *B* aus sowohl die Objektart *A* als auch die Objektart *A_Symbol* (d.h. der konzeptionelle Schlüssel) referenziert werden. Die Relationen, die von *A* ausgehen, verbleiben dagegen bei *A* (beispielsweise die Relation von *A* zu *B*). Da bei dieser Konstruktion auf der Seite von *A* eine Aufteilung der Relation auf die beiden Objektarten *A* und *A_abstrakt* vorgenommen wird, gibt es hier nicht eine primäre und eine dazu inverse Relation, sondern zwei primäre Relationen (ohne inverse Relationen). Aus diesem Grund bleiben im NIAM-Diagramm die entsprechenden Rollenbeschreibungen bei den Relationen leer:



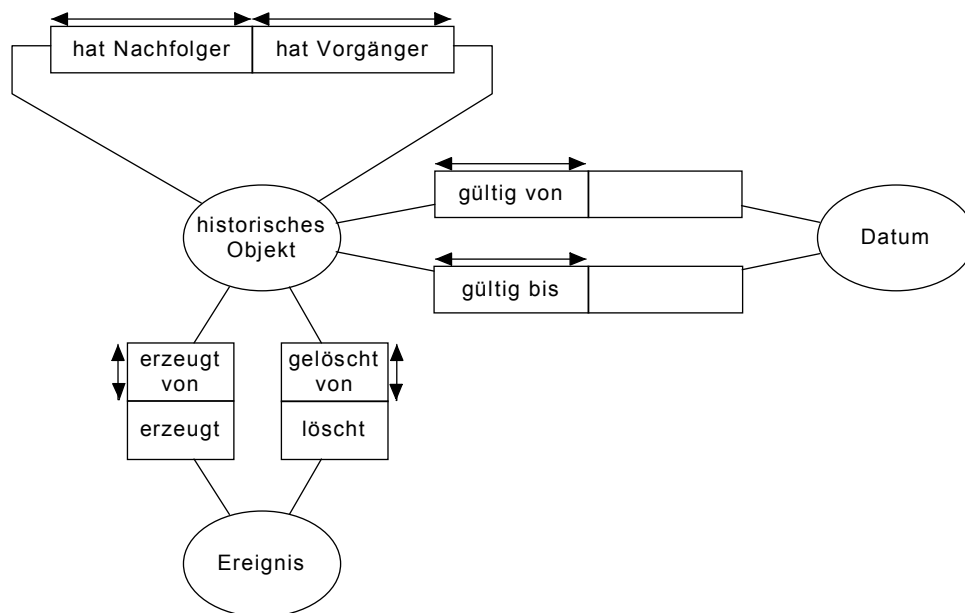
Es ist auch möglich, sowohl die Relation von *B* nach *A* als auch diejenige von *A* nach *B* in Form eines abstrakten Verweises zu formulieren. Auch dabei entstehen zwei primäre Relationen ohne die entsprechenden inversen Relationen:



Abstrakte Verweise sind grundsätzlich nur bei solchen Objektarten möglich, für die sich ein geeigneter konzeptioneller Schlüssel in Form eines eindeutigen Ordnungsmerkmals finden lässt (Beispiele: Netzknotennummer, Flurstückskennzeichen etc.). Der Aufbau des konzeptionellen Schlüssels wird im zugehörigen NIAM-Dokument beschrieben. Da das derzeitige Konzept für Objekt_IDs im OKSTRA® nur optionale IDs vorsieht, kommen diese als konzeptionelle Schlüssel nicht in Betracht.

Historisierung

Der OKSTRA® besitzt ein Konzept zur Historisierung von Daten auf Objektebene. Durch Erben von der Objektart *historisches Objekt* aus dem Schema „Historisierung“ erhält eine Objektart die Fähigkeit zur Nutzung dieses Konzeptes. Eine Instanz einer solchen Objektart ist als Objektversion zu interpretieren, die nur für ein bestimmtes Zeitintervall gültig ist. Von einem Objekt können damit mehrere Objektversionen existieren, die die verschiedenen Zustände des Objektes im Verlauf der Zeit angeben. Die einzelnen Objektversionen können (und sollten) über Vorgänger- und der Nachfolgerrelationen miteinander verknüpft werden. Es besteht auch die Möglichkeit, ein *Ereignis* anzugeben, das für den Sprung von einer Objektversion zur nächsten verantwortlich ist.



Das Gültigkeitsintervall wird durch die beiden Attribute „gültig von“ und „gültig bis“ realisiert. Durch die Verwendung des Types *Datum* erfolgen diese Angaben tagesscharf; dies ist so zu interpretieren, dass eine Objektversion von 0:00 Uhr des beim Attribut „gültig von“ angegebenen Datums bis 24:00 Uhr des beim Attribut „gültig bis“ angegebenen Datums gültig ist. Bei einer noch nicht abgeschlossen Version, die „bis auf weiteres“ gültig ist, wird für das Attribut „gültig bis“ kein *Datum* angegeben.

Die einzelnen Versionen eines Objektes können sich über die Relationen „hat Vorgänger“ und „hat Nachfolger“ gegenseitig referenzieren. Mit diesen Relationen lässt sich eine doppelt verkettete Liste aller Versionen zu einem Objekt bilden, wobei bei der ersten Version die Relation „hat Vorgänger“ und bei der letzten Version die Relation „hat Nachfolger“ nicht belegt ist:



Für die einzelnen Objektversionen innerhalb der Kette gelten folgende Konsistenzbedingungen:

1. Das „gültig bis“-Datum einer Version darf nicht vor dem „gültig von“-Datum derselben Version liegen. Es darf allerdings fehlen, wenn die Version noch nicht beendet ist. Sind beide Angaben gleich, ist die Version von 0:00 Uhr bis 24:00 Uhr des angegebenen Tages gültig.
2. Das „gültig von“-Datum einer Version muss dem auf das „gültig bis“-Datum der Vorgängerversion folgenden Tag entsprechen. Ausgenommen hiervon ist lediglich die erste Objektversion innerhalb der Kette, weil sie keine Vorgängerversion besitzt.

Die Objektart *Ereignis* ist momentan sehr straßenbaubezogen modelliert und kann daher nur bei Straßenbauvorhaben sinnvoll eingesetzt werden. Beispielsweise müssen verpflichtend die PROJIS-Nummer, die Verkehrseinheit und die *Straßenbaudienststellen* angegeben werden, in deren Bereich die Baumaßnahme erfolgt.

Da bei einer historischen Objektart semantisch betrachtet erst alle Versionen eines Objektes zusammen die komplette Objektinstanz realisieren, werden alle Relationen, die auf historische Objektarten verweisen, im OKSTRA[®]-EXPRESS-Schema multipel, auch dann, wenn sie konzeptionell eindeutig sind. In den NIAM-Diagrammen wird dieser technische Aspekt nicht berücksichtigt, d.h. dort ist eine konzeptionell eindeutige Relation zu einem historischen Objekt im Unterschied zum EXPRESS-Schema auch als eindeutige Relation angegeben. Um eine Relation zu einem historischen Objektes auswerten zu können, muss ein Zeitpunkt angegeben werden, für den die Auswertung erfolgen soll. Aus der Menge der insgesamt referenzierten Objektversionen können dann diejenigen herausgesucht werden, die zum angegebenen Zeitpunkt gültig sind.

Sofern für ein historisches Objekt eine ID vergeben wird, gilt diese für alle Versionen des Objektes, da sich zwar von einer Version zur nächsten möglicherweise einige Eigenschaften ändern, nicht aber die grundsätzliche Identität des Objektes. Bei historischen Objekten ist daher nicht die ID allein, sondern erst die Kombination von ID und Zeitpunkt eindeutig.

Um das Historisierungskonzept des OKSTRA[®] sinnvoll nutzen zu können, sollte für die betroffenen Objektarten individuell festgelegt werden, welche Eigenschaftsänderungen lediglich versionsbildend sind, d.h. zur Erzeugung einer neuen Objektversion führen, und welche objektbildend sind, also zum kompletten Untergang eines Objektes und zur Erzeugung eines neuen Objektes (mit einer eigenen Kette von Objektversionen) führen. Dies leistet der OKSTRA[®] momentan nicht; hier müssen bei Bedarf in den einzelnen Fachsystemen entsprechende Regeln erstellt werden.

Es sei noch darauf hingewiesen, dass sich bestimmte Historisierungsinformationen mit dem im OKSTRA[®] verwendeten Modell nicht abbilden lassen: Situationen, bei denen aus einer Objektversion mehrere neue entstehen oder bei denen aus mehreren Versionen eine einzige neue Version erzeugt wird, lassen sich mit dem OKSTRA[®]-Historisierungskonzept nicht abbilden. Es ist nicht möglich, Vorgänger- und Nachfolgerrelationen auf Objekte mit anderer Identität aufzubauen. Außerdem ist es nicht möglich, Änderungen an Teilen geometrischer Objekte explizit anzugeben (nach dem Historisierungs-Modell des OKSTRA[®] muss sich entweder das gesamte Objekt ändern oder gar nichts).

Aus diesem Grund gibt es teilweise einen Bedarf an objektartspezifischen Erweiterungen des Historisierungs-Modells: Beispielsweise existiert für die Historisierung von *Abschnitten* und *Ästen* im Straßennetz die Objektart *identisches_Netzteil*, mit der angegeben werden kann, dass ein Teil eines *Abschnittes* oder *Astes* gleichgeblieben ist, obwohl der gesamte *Abschnitt* oder *Ast* versioniert (oder sogar durch ein Objekt mit neuer Identität) ersetzt worden ist. Dies wird dadurch erreicht, dass zwei *Teilabschnitte*, einer auf dem alten und einer auf dem neuen Straßenverlauf, über eine Instanz der Objektart *identisches_Netzteil* miteinander gekoppelt werden.

Geometrie

Das Geometrieschema des OKSTRA® stellt vier abstrakte Supertypes bereit, mit denen einer Objektart Koordinatengeometrie (und/oder ein topologischer Bezug) zugeordnet werden kann:

1. Erbt eine Objektart vom *Punktobjekt_Modell*, kann sie (optional) eine Punktgeometrie tragen.
2. Erbt eine Objektart vom *Linienobjekt_Modell*, kann sie (optional) eine Liniengeometrie tragen.
3. Erbt eine Objektart vom *Flächenobjekt_Modell*, kann sie (optional) eine Flächengeometrie tragen.
4. Erbt eine Objektart vom *Volumenobjekt_Modell*, kann sie (optional) eine Volumengeometrie tragen (einen von Flächen begrenzten Raumkörper).

Eine einzelne Koordinate einer solchen Geometrie wird über die Objektart *direct_position* aus dem Schema „direct_positioning_schema“ abgebildet und ist zwei- oder dreidimensional. Außerdem können optional ein Referenzsystem (Attribut „in_pos_ref_system“) und ein Koordinatensystem (Attribut „in_coordinate_system“) angegeben werden. Das Attribut „In_pos_ref_system“ gibt das verwendete Geodätische Referenzsystem an, z.B. das „Deutsche Hauptdreiecksnetz“ oder „WGS84“. Das Attribut „in_coordinate_system“ beschreibt die verwendete kartographische Abbildung, z.B. „3-Grad-Gauß-Krüger-Streifen, 3. Meridianstreifen“. Welche Referenzsysteme und Koordinatensysteme zulässig sind, ist im Dokument D018 zu finden.

Straßennetzbezug von Objekten

Das Schema „Straßennetz“ stellt abstrakte Supertypen für Objektarten bereit, die sich auf das ASB-Straßennetz beziehen. Möglich sind Punkt-, Strecken- und Bereichsbezüge. Es gibt darüber hinaus abstrakte Supertypen, die neben dem Straßennetzbezug gleichzeitig die Fähigkeit zur Historisierung vererben. Insgesamt existieren folgende Supertypen:

Abstrakter Supertyp	Verortung im Straßennetz	<i>historisches Objekt</i>
<i>Punktobjekt_stat</i>	als <i>Straßenpunkt</i>	nein
<i>Punktobjekt_hist</i>	als <i>Straßenpunkt</i>	ja
<i>Streckenobjekt_stat</i>	als <i>verallgemeinerte Strecke</i>	nein
<i>Streckenobjekt_hist</i>	als <i>verallgemeinerte Strecke</i>	ja
<i>Bereichsobjekt_stat</i>	als <i>Netzbereich</i>	nein
<i>Bereichsobjekt_hist</i>	als <i>Netzbereich</i>	ja

Grundlegende Typdefinitionen

Das Schema „Allgemeine Objekte“ stellt grundlegende Typdefinition für Attribute zur Verwendung in anderen Schemata bereit, beispielsweise die Datentypen *Meter*, *Stundenkilometer*, *Winkel*, *Prozent* und *Tonnen*.