

Dieses Dokument erläutert die gewählte Vorgehensweise zur Modellierung der Historisierung im OKSTRA und diskutiert ihre Vor- und Nachteile.

In Abschnitt 1.1 wird die Zielsetzung einer Historisierung im OKSTRA kurz erläutert.

In Abschnitt 1.2 wird ein NIAM-Schema für die Historisierung dargestellt und erläutert.

In Abschnitt 1.3 wird die Integration der gewählten Variante der Historisierung in das bisherige Datenschema des OKSTRA diskutiert.

In Abschnitt 1.4 wird eine mögliche Modellierung in EXPRESS angegeben und erläutert.

In Abschnitt 1.5 werden kurz die Auswirkungen der Historisierung auf das SQL-Schema des OKSTRA diskutiert.

In Abschnitt 1.6 werden alternative Verfahren zur Historisierung analysiert.

## **1.1 Zielsetzung**

In einem Workshop, an dem Vertreter der einzelnen Teilprojekte sowie der BG OKSTRA teilgenommen haben, wurden die Anforderungen an eine Historisierung für den OKSTRA festgelegt. Die Ergebnisse dieses Workshops liegen diesem Dokument zugrunde.

Der OKSTRA soll die Fähigkeit besitzen, Objektzustände im Verlauf der Zeit zu erfassen. Das heißt, er soll die Möglichkeit bieten, Situationen zu rekonstruieren, wie sie an einem gegebenen Zeitpunkt vorlagen. Insbesondere muß das Straßennetz zu jedem beliebigen Zeitpunkt verfügbar sein, da angegliederte Daten sich im allgemeinen auf den Zustand des Straßennetzes zum Zeitpunkt der Erfassung dieser Daten beziehen. In den Teilprojekten wurde mit Hilfe der zuständigen Experten entschieden, welche Objekte historisiert werden müssen.

Es ist zu beachten, daß die Historisierung mehr als nur eine weitere Eigenschaft einiger Objekte ist. Vielmehr erhält der OKSTRA eine zusätzliche Zeitdimension, die orthogonal zu allen übrigen Informationen steht. Die Komplexität einer möglichen Implementation steigt durch die Integration einer Historisierung erheblich.

## **1.2 Datenschema zur Historisierung**

Nachfolgend ist ein NIAM-Schema zur Historisierung dargestellt. Es ist aus dem entsprechenden Diagramm aus dem Teilprojekt 1 abgeleitet, wurde jedoch etwas modifiziert. Dies ist zulässig, da das ursprüngliche Diagramm in gewisser Weise als Modellerweiterung des OKSTRA-Datenmodells verstanden werden kann und muß.

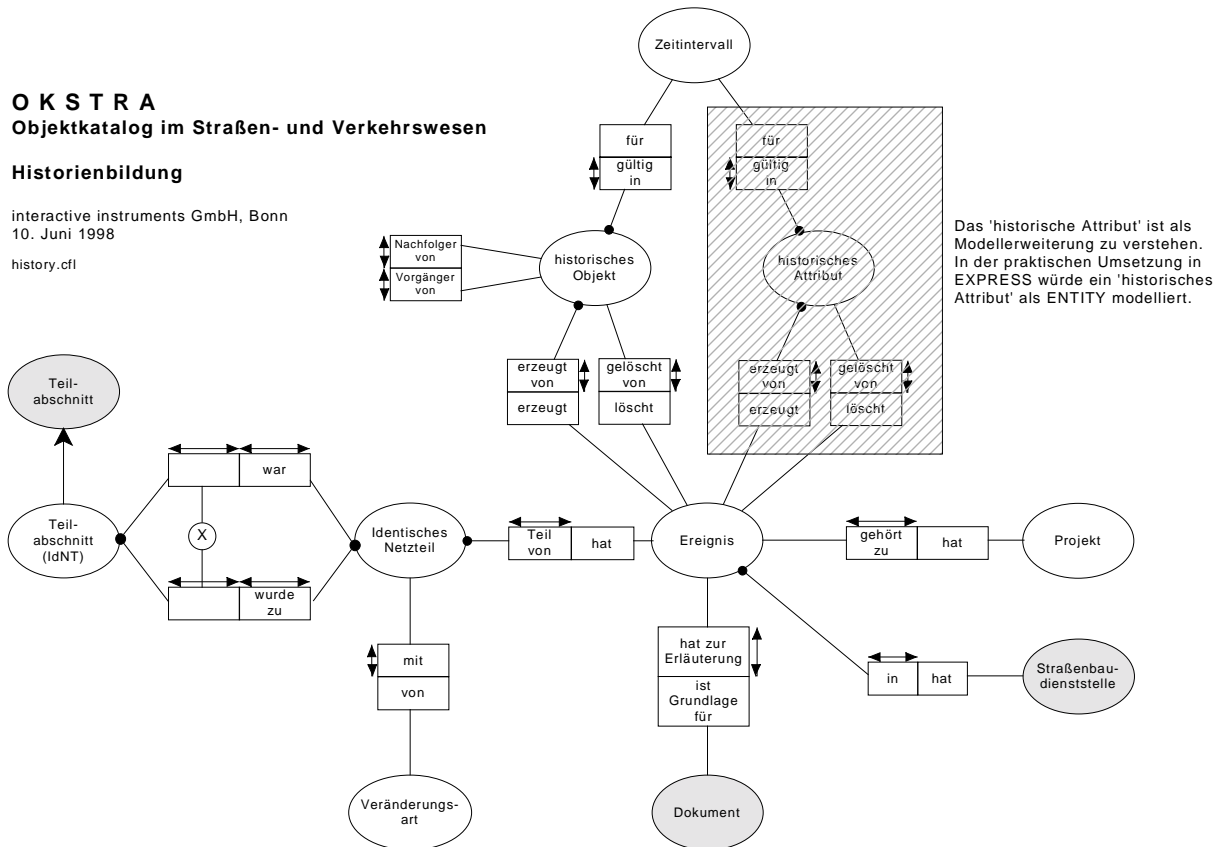


**OKSTRA**  
Objektkatalog im Straßen- und Verkehrswesen

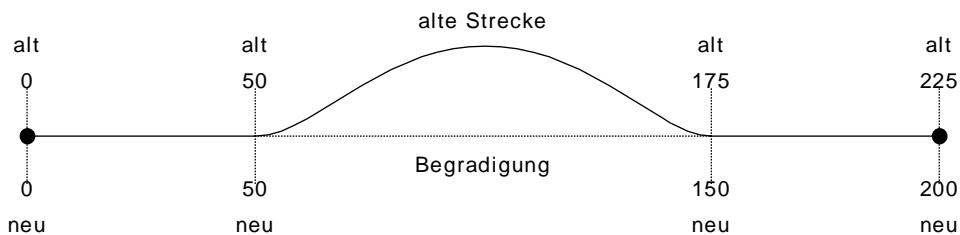
**Historienbildung**

interactive instruments GmbH, Bonn  
10. Juni 1998

history.cfl



Zur Erläuterung: Eine Objektklasse, die historisiert werden soll, erbt aus dem 'historischen Objekt'. Dadurch kann einem Objekt ein 'Zeitintervall', in dem es gültig ist, zugeordnet werden, und eine Chronologie von Vorgänger und Nachfolger kann gespeichert werden. Eine spezielle Modellierung ist im Kernbereich des Straßennetzes notwendig, um die historische Entwicklung eines realen Straßenstücks zu speichern. Betrachte dazu folgendes Diagramm:

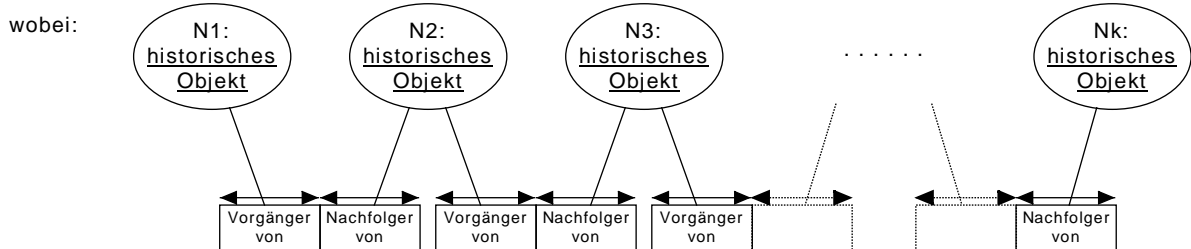
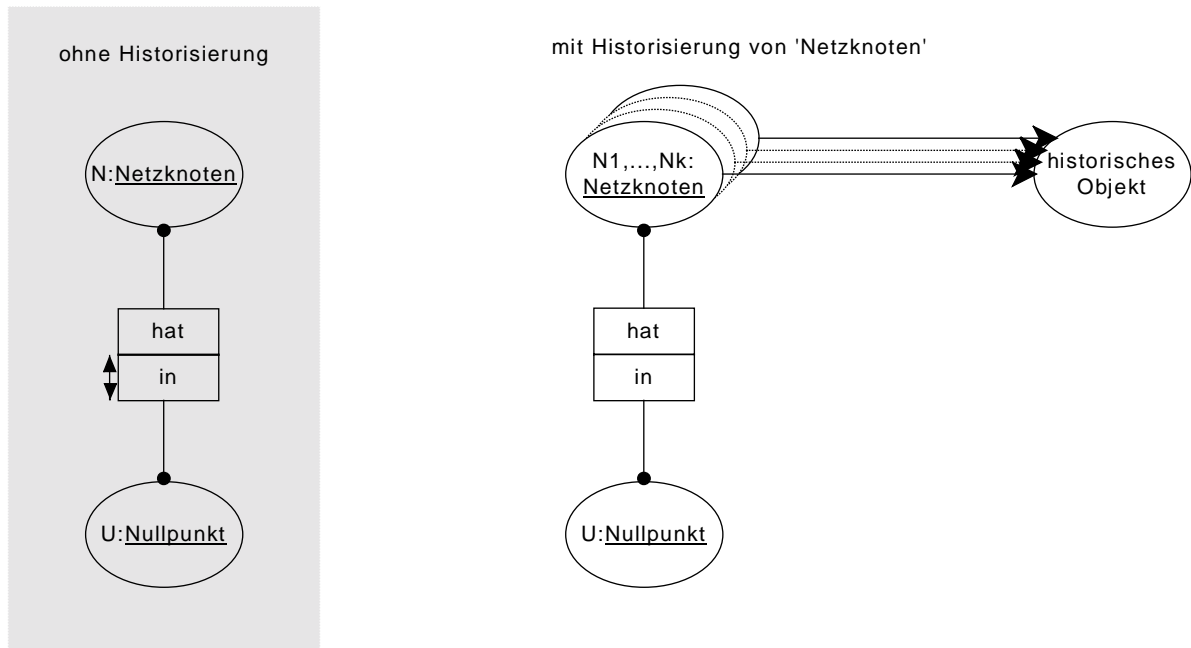


Der Teilabschnitt von Station 150(neu) bis 200(neu) des neuen Abschnitts ist *identisch* mit dem Teilabschnitt von Station 175(alt) bis 225(alt) des alten Abschnitts. Diese Beziehung wird durch die Relationen des Objekts 'Identisches Netzteil' zu 'Teilabschnitt (IdNT)' gespeichert.



### 1.3 Integration in das Datenschema des OKSTRA

An einem einfachen Beispiel sollen zunächst die grundlegenden Auswirkungen der Historisierung bei der vorgesehenen Modellierung erläutert werden. Betrachte dazu folgendes, an die NIAM-Syntax angelehntes Diagramm (dabei bezeichnet z.B. 'Netzknoten' ein Objekt der Objektklasse 'Netzknoten')



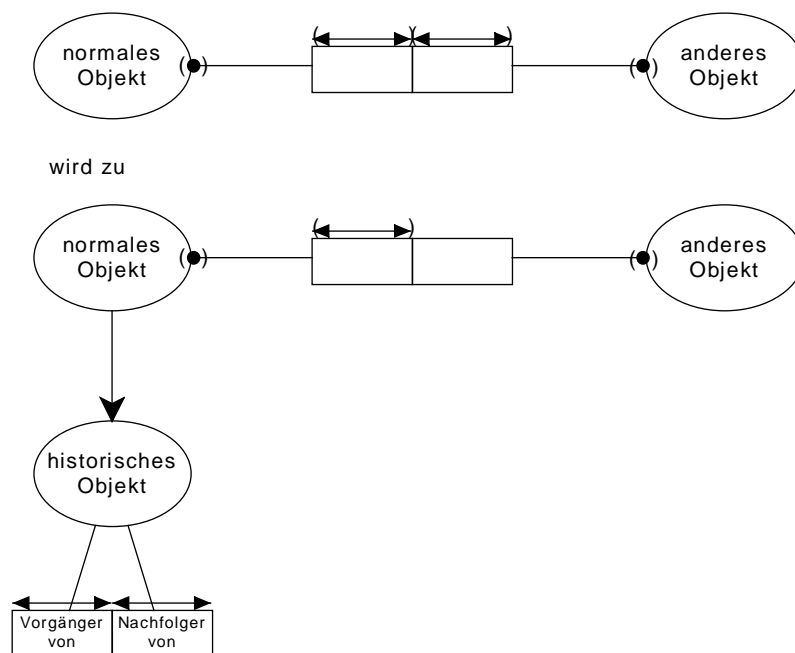
Ohne Historisierung gehört ein 'Nullpunkt' zu genau einem 'Netzknoten'. Wird der 'Netzknoten' jedoch historisiert, so erbt der 'Netzknoten' aus dem 'historischen Objekt' und ein 'Nullpunkt' gehört dann zu einer Kette von 'Netzknoten', die den aufeinanderfolgenden Zuständen des vormals eindeutigen 'Netzknotens' entsprechen. Dabei wird die Reihenfolge der historisierten 'Netzknoten' durch Relationen der zugehörigen Instantisierungen ihres Supertypes 'historisches Objekt' gespeichert. Das 'historische Objekt' achtet dann sozusagen selbst darauf, daß die zugehörigen Gültigkeitszeiträume konsistent sind.

Die Handhabung 'historischer Attribute' verläuft ganz ähnlich (wobei momentan keines der drei Teilprojekte die Möglichkeit historisierender Attribute nutzt). Praktisch würde ein solches Attribut als eine historisierende Objektklasse mit diesem Attribut modelliert.



Wie in obigem Beispiel zu erkennen, hat die Historisierung Einfluß auf die Relationen zu einer historisierenden Objektklasse hin. Generell wird eine Relation zu einer historisierenden Objektklasse multipel, da aus jedem einzelnen Objekt durch die historische Entwicklung eine Menge von Objekten wird. Das bedeutet anschaulich: auf Relationsebene fallen eventuell vorhandene Pfeile über den entsprechenden Relationskästchen weg. Damit geht Struktur im Datenschema verloren, da dem Schema nicht mehr anzusehen ist, ob es sich zu einem gewählten Zeitpunkt um eine eindeutige oder mehrdeutige Relation handelt. Gewissermaßen vermischt diese Vorgehensweise die Kardinalität einer Relation zu einem festen Zeitpunkt und die zeitliche Abfolge.

Diese Auswirkung soll noch einmal an einem Diagramm verdeutlicht werden:



Letztlich kann aber nur eine Implementation des OKSTRA sicherstellen, daß die Kardinalitäten der Relationen korrekt sind, d.h. etwa eine vormals eindeutige Relation nach Berücksichtigung der Historisierung nur jeweils ein Objekt zu jedem Zeitpunkt zuläßt. Einige Konsistenzbedingungen, die vorher durch das (EXPRESS-)Datenschema geprüft wurden, können nach Integration der Historisierung nicht mehr auf diese Weise geprüft werden.

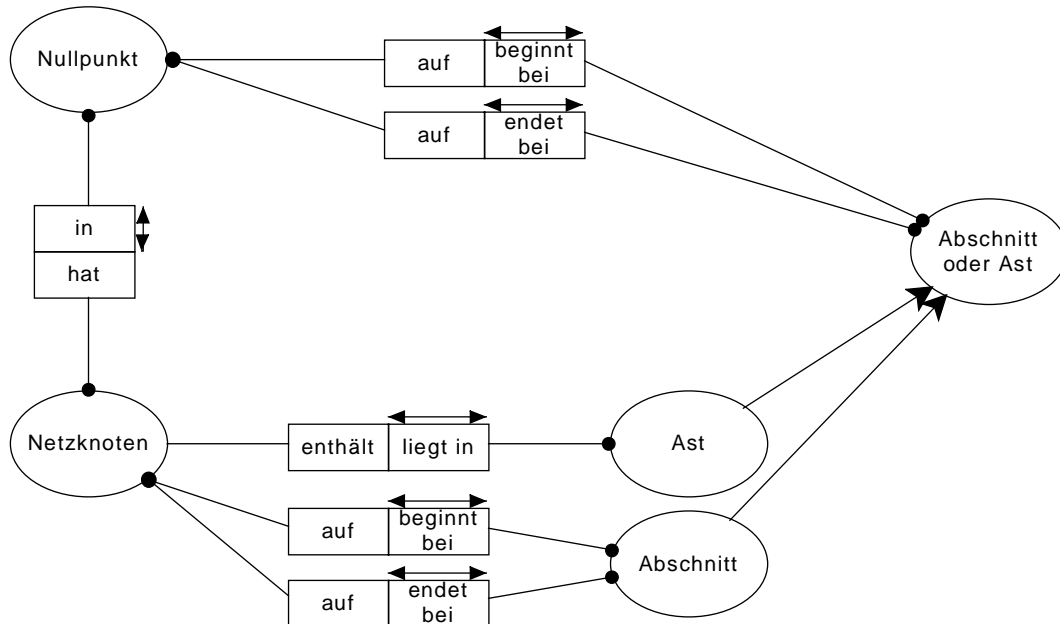
Es erscheint sinnvoll, Objekte ohne hinreichendes Eigenleben von einer getrennten Historisierung auszunehmen, etwa 'Straßenpunkte' und 'Teilabschnitte'. Diese sollten eher als Eigenschaften derjenigen Objekte aufgefaßt werden, die sie referenzieren. Bei einer Änderung der Eigenschaft würden die referenzierenden Objekte historisiert. Die beiden genannten Objekte würden dann eine feste Instanz eines 'Abschnitt oder Asts' referenzieren und mit der Historisierung dieser Instanz ihre Gültigkeit ebenfalls verlieren.

Die beiden folgenden Diagramme stellen einen vereinfachten Ausschnitt aus dem Datenschema des Straßennetzes dar. Anhand der Relationen zwischen 'Netzknoten', 'Null-



punkt', 'Abschnitt' und 'Ast' ohne bzw. mit Integration der Historisierung soll ein Eindruck von den Auswirkungen der Historisierung auf das Datenschema gegeben werden<sup>1</sup>:

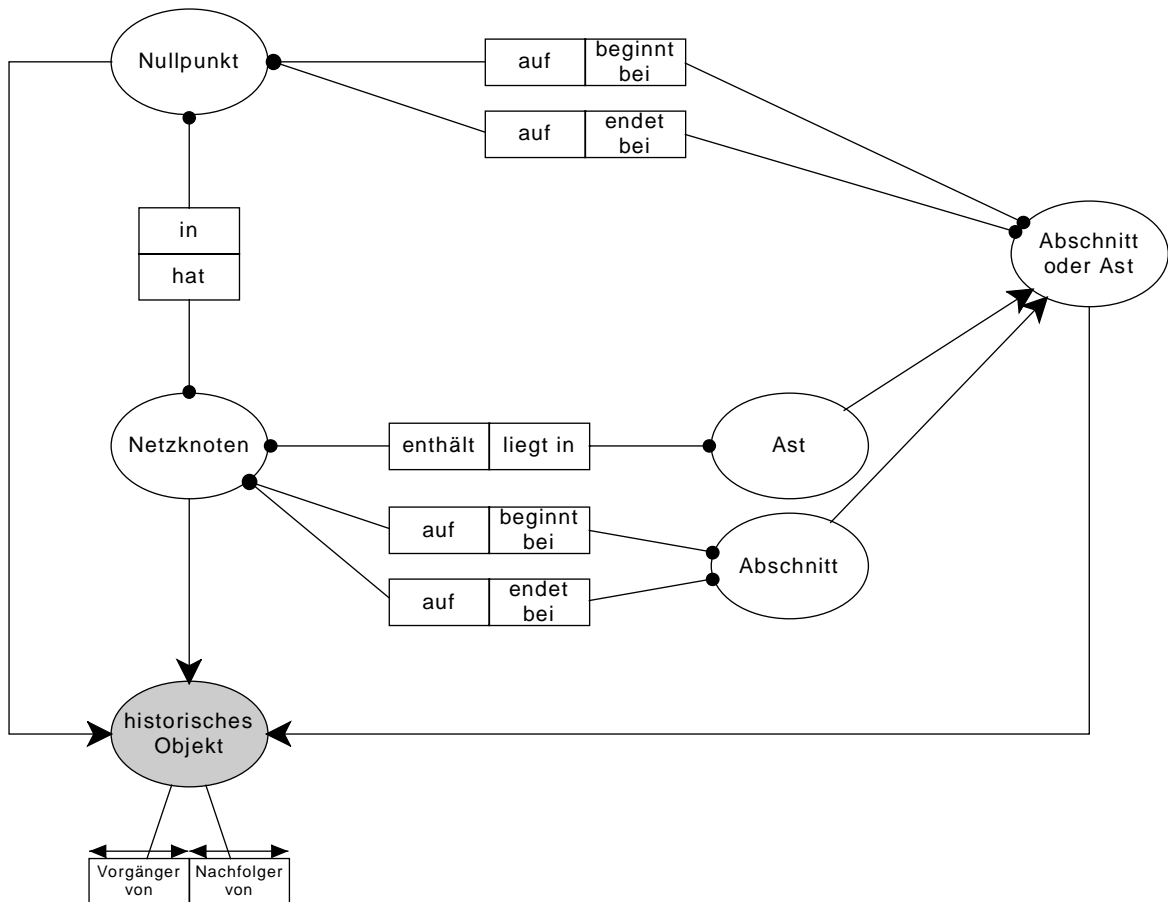
- ohne Historisierung:



<sup>1</sup> Auf eine Integration der Historisierung in die NIAM-Diagramme des OKSTRA-Datenschemas wurde bewusst verzichtet, um die Aussagekraft der Diagramme nicht herabzusetzen. Hier soll nur beispielhaft demonstriert werden, wie sich die tatsächlichen Kardinalitäten durch Integration der Historisierung verändern.



- mit Historisierung:



## 1.4 Modellierung in EXPRESS

Bei der Modellierung der Historisierung wurde versucht, die Auswirkungen auf das übrige Datenschema möglichst gering zu halten. Folgende Auswirkungen sind jedoch anzumerken:

- Wie oben erläutert wird die Historisierung durch Vererbung aus dem 'historischen Objekt' und eine Flexibilisierung sämtlicher Relationen zu einer historisierenden Objektklasse hin, wobei jede solche Relation zu einer SET-Relation wird, in das EXPRESS-Datenschema des OKSTRA integriert.
- DERIVE-, UNIQUE- und WHERE-Clauses in historisierenden Objektklassen werden an vielen Stellen komplexer oder sind überhaupt nicht mehr sinnvoll umsetzbar, da sie zeitabhängig werden.

Nachfolgend ist eine Modellierung des Datenschemas zur Historisierung im OKSTRA in EXPRESS gegeben (die Liste der SUBTYPEs im Kopf des ENTITYs 'historisches\_Objekt' wurde der Übersichtlichkeit halber durch '...' ersetzt, auf die entsprechenden REFERENCE-Verweise auf andere SCHEMAS wurde aus dem gleichen Grund verzichtet):



```
SCHEMA Historisierung;

REFERENCE FROM Datentypen (Datum);

REFERENCE FROM Strassennetz (Teilabschnitt);

REFERENCE FROM Administration (Dokument,Strassenbaudienststelle);

ENTITY historisches_Objekt
ABSTRACT SUPERTYPE OF (ONEOF(...));
  (* ARTEN J *)
  (* MIX_IN gueltig_von,gueltig_bis *)
--- Attribute :
  gueltig_von          : OPTIONAL Datum;
  gueltig_bis         : OPTIONAL Datum;
--- Relationen :
  erzeugt_von_Ereignis : Ereignis;
  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;
LOCAL
  NJ,NM,NT,VJ,VM,VT      : INTEGER;
  VE,NA                  : Datum;
  VO                     : historisches_Objekt;
END_LOCAL;
  IF EXISTS(h0.hat_Vorgaenger_hist_Objekt) THEN
    VO := h0.hat_Vorgaenger_hist_Objekt;
    IF EXISTS(h0.gueltig_von) AND EXISTS(VO.gueltig_bis) THEN
      NA := h0.gueltig_von;      -- neues Objekt / Nachfolger
      NJ := VALUE(NA[7:10]);     -- Jahr des Enddatums
      NM := VALUE(NA[4:5]);     -- Monat des Enddatums
      NT := VALUE(NA[1:2]);     -- Tag des Enddatums
      VE := VO.gueltig_bis;     -- altes Objekt / Vorgaenger
      VJ := VALUE(VE[7:10]);    -- Jahr des Anfangsdatums
      VM := VALUE(VE[4:5]);    -- Monat des Anfangsdatums
      VT := VALUE(VE[1:2]);    -- Tag des Anfangsdatums
      RETURN ( ( NJ * 10000 + NM * 100 + NT )
                > ( VJ * 10000 + VM * 100 + VT ) );
    ELSE
      RETURN (FALSE);
    END_IF;
  ELSE
    RETURN (TRUE);
  END_IF;
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;

ENTITY Projekt_Strassenbau;
--- Attribute :
--- Relationen :
INVERSE
  hat_Ereignis          : SET [0:?] OF Ereignis
                        FOR gehoert_zu_Projekt;
END_ENTITY;

ENTITY identisches_Netzteil;
--- Attribute :
--- Relationen :
  mit_Veraenderungsart : OPTIONAL Veraenderungsart;
  war_Teilabschnitt    : OPTIONAL Teilabschnitt_IdNT;
  wurde_zu_Teilabschnitt : OPTIONAL Teilabschnitt_IdNT;
INVERSE
  Teil_von_Ereignis    : Ereignis FOR hat_identisches_Netzteil;
WHERE
  Bezug_zu_Teilabschnitt : EXISTS(war_Teilabschnitt) OR
                        EXISTS(wurde_zu_Teilabschnitt);
  (* BEDINGUNG ( war_Teilabschnitt IS NOT NULL ) OR
    ( wurde_zu_Teilabschnitt IS NOT NULL ) *)
END_ENTITY;

ENTITY Veraenderungsart;
  (* KEY_NAME Kennung *)
  (* KEY_TYP CHAR(2) *)
  Kennung          : STRING(2) FIXED;
  Langtext         : STRING;
UNIQUE
  Kennung_eindeutig : Kennung;
END_ENTITY;

(* SQL :

INSERT INTO Veraenderungsart VALUES ('NB','Neubau')
INSERT INTO Veraenderungsart VALUES ('VB','Vollausbau')
INSERT INTO Veraenderungsart VALUES ('ZB','Zwischenausbau')
INSERT INTO Veraenderungsart VALUES ('RE','Rekultivierung')
INSERT INTO Veraenderungsart VALUES ('DT','Deckenerneuerung im Tiefeinbau')
INSERT INTO Veraenderungsart VALUES ('DH','Deckenerneuerung im Hocheinbau')
INSERT INTO Veraenderungsart VALUES ('WI','Widmung')
INSERT INTO Veraenderungsart VALUES ('AU','Aufstufung')
INSERT INTO Veraenderungsart VALUES ('AB','Abstufung')
INSERT INTO Veraenderungsart VALUES ('UM','Umnummerierung in der ' ||
'Straßenbezeichnung')
INSERT INTO Veraenderungsart VALUES ('EZ','Einziehung')
INSERT INTO Veraenderungsart VALUES ('OD','Verlegung der OD-Grenze')
INSERT INTO Veraenderungsart VALUES ('UI','UI-Vertrag')
INSERT INTO Veraenderungsart VALUES ('GR','Grenzverlegung')
INSERT INTO Veraenderungsart VALUES ('SB','Änderung der Abschnitts-' ||
'Astbezeichnung')
INSERT INTO Veraenderungsart VALUES ('LN','Längenänderung infolge Neumessung')
INSERT INTO Veraenderungsart VALUES ('KO','Korrektur')

  END_SQL
*)

ENTITY Teilabschnitt_IdNT
SUBTYPE OF (Teilabschnitt);
  (* EINMAL Abschnitt_oder_Ast *)
--- Attribute :
--- Relationen :
INVERSE
  altes_Netzteil : SET [0:1] OF identisches_Netzteil
                FOR war_Teilabschnitt;
  neues_Netzteil : SET [0:1] OF identisches_Netzteil
```





```
FOR wurde_zu_Teilabschnitt;  
WHERE  
    entweder_alt_oder_neu      : EXISTS(altes_Netzteil) XOR  
                                EXISTS(neues_Netzteil);  
END_ENTITY;  
END_SCHEMA; -- Historisierung
```

Abgesehen vom 'historischen Objekt' ist dies eine direkte Umsetzung des NIAM-Schemas. Das Objekt 'Zeitintervall' wurde in je ein optionales Attribut vom Typ 'Datum' für den Beginn und eins für das Ende des Gültigkeitsintervalls aufgespalten.

Das 'historische Objekt' prüft automatisch die zeitliche Konsistenz einer Kette 'historischer Objekte' mittels der selbstdefinierten EXPRESS-FUNCTION 'Objektfolge\_konsistent'. Hat ein 'historisches Objekt' keinen Vorgänger, so ist nichts zu prüfen. Existiert ein Vorgänger, so wird vom Datenbestand verlangt, daß dieser Vorgänger ein Datum für das Ende seines Gültigkeitszeitraums trägt, der Nachfolger eines für den Beginn, und daß letzteres zeitlich nach ersterem liegt. Diese Konsistenzbedingung wird für jedes Objekt geprüft. Folglich wird vermöge der Transitivität der Bedingungen die Gültigkeit der Zeitbedingung für die ganze Kette von 'historischen Objekten' geprüft.

### 1.5 Auswirkungen auf das SQL-Schema des OKSTRA

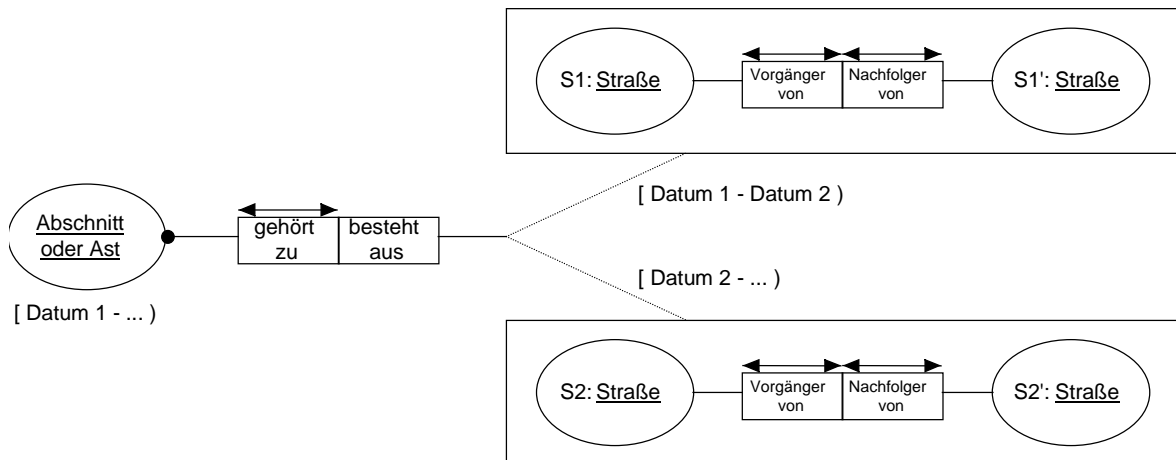
An den grundsätzlichen Modellierungsmethoden für das SQL-Schema des OKSTRA ändert die Integration der Historisierung nichts. Folgende Auswirkungen sind dennoch anzumerken:

- Durch die Auflösung von Eindeutigkeiten bei den Relationen zu historisierenden Objektklassen hin wird eine Vielzahl zusätzlicher TABLEs für Relationen notwendig.
- Grundsätzlich ist für eine Abfrage an eine OKSTRA-Datenbank die zeitliche Gültigkeit der Information einzubeziehen. Um die dafür notwendige Operation zu beschleunigen, werden die Attribute 'gueltig\_von' und 'gueltig\_bis' jeweils direkt in die darstellende TABLE eines historisierenden Objekts eingetragen.

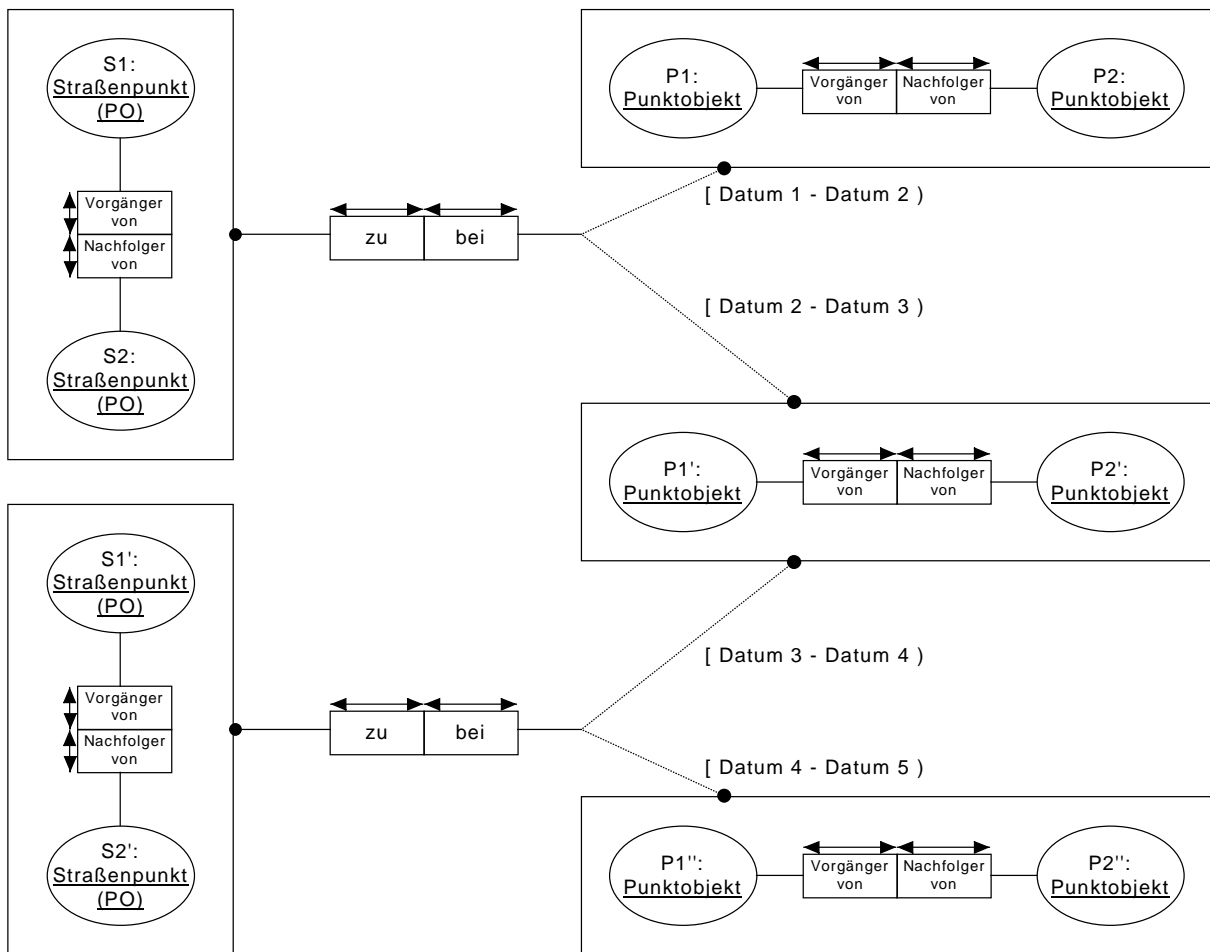
### 1.6 Alternativen zum vorgestellten Verfahren zur Historisierung

Eine Funktion der Historisierung ist die Möglichkeit, die verschiedenen Zustände eines Objekts im OKSTRA im zeitlichen Verlauf zu dokumentieren. Beispiel: einer 'Straße' wird ein weiterer 'Abschnitt oder Ast' zugeordnet. Diese veränderte 'Straße' übernimmt dann die Rolle ihrer Vorgängerin und steht mit dieser über die 'Vorgänger-Nachfolger'-Relation der zugehörigen 'historischen Objekte' in Verbindung. Auf diese Art und Weise entstehen die zu Beginn des Abschnitts 1.3 beschriebenen Ketten.

Auf den ersten Blick wäre es naheliegend, diese Ketten zu einer neuen Objektklasse, gewissermaßen einem 'Straßen-Container', zusammenzufassen. Damit würde der historische Zusammenhang der enthaltenen 'Straßen' deutlicher hervorgehoben. Ziel eines solchen Vorgehens wäre es, die Historisierung stärker auf die Objektebene zu verlagern und die Relationen, insbesondere in ihren Kardinalitäten, möglichst unverändert zu lassen. Es stellt sich jedoch heraus, daß dieses Vorgehen problematisch ist. An einem einfachen Beispiel soll dies erläutert werden. Ein 'Abschnitt oder Ast' gehört zu genau einer 'Straße'. In folgendem Szenario wird diese Zuordnung am Datum 2 geändert:



Das bedeutet, abhängig vom gewählten Zeitpunkt müßte die eindeutige Relation 'gehört zu' von dem 'Abschnitt oder Ast' entweder zum ersten oder zum zweiten 'Straßen-Container' gehen. Eine solche 'temporäre Relation' ist im gegebenen Rahmen nicht modellierbar. Um diese Freiheit zu gewinnen wäre es notwendig, Relationen in EXPRESS als eigene Objekte zu modellieren. Dann jedoch sollte, um eine unangemessene Komplexität des EXPRESS-Schemas des OKSTRA zu vermeiden, eine gegebene Relation zwischen zwei Objektklassen durch ein einziges Objekt dargestellt werden. Aber auch dieses ist nicht mit den gegebenen Mitteln modellierbar. Anhand des einfachen Falls einer beidseitig eindeutigen Relation soll diese Problematik erläutert werden: Angenommen, die gegebenen 'Punktobjekte' sind 'Betriebskilometer'. Bei Datum 2 bzw. 4 könnte etwa die Blockeinteilung geändert werden, so daß an der gleichen Stelle ein völlig anderer 'Betriebskilometer' läge. Bei Datum 3 könnte sich infolge einer neuen Anschlußstelle durch Aufteilung in zwei 'Abschnitte' der zugehörige 'Abschnitt' ändern, so daß ein anderer 'Straßenpunkt' an derselben Stelle läge.



Um die Kardinalitäten der Relationen zwischen 'Straßenpunkt' und 'Punktobjekt' zu bewahren, müßte hier ein eindeutiges Relationsobjekt sowohl die Relation des oberen 'Straßenpunkts (PO)' zum 'Punktobjekt' als auch die umgekehrte Richtung darstellen. Ändern sich nun im Verlauf der Zeit auf beiden Seiten die zugehörigen Relationspartner, so läßt sich dieses nicht mehr eindeutig mit einem einzigen Objekt darstellen.

Eine Lösung dieses Problems würde erfordern, die beiden Seiten einer Relation zu trennen und einzeln zu verwalten. Damit würde die Komplexität des EXPRESS-Schemas allerdings erheblich zunehmen und der praktische Nutzen einer solchen Lösung wäre sehr fraglich. Im SQL-Schema würde die Anzahl der Tabellen noch einmal drastisch steigen. Darüber hinaus würde dann die Bündelung der Objekte in Ketten gemäß ihrer historischen Abfolge durch eine Aufspaltung in den Relationen größtenteils wieder rückgängig gemacht.

Eine sinnvolle Lösung, die die Identität von Objekten stärker bewahrt, wie dies etwa durch das Bilden der Container-Objekte bezweckt wird, würde eine Berücksichtigung entsprechender zeitlicher Abhängigkeiten bereits in der Modellersprache bzw. dem verwendeten Datenbanksystem erfordern. Derartige Ansätze sind zur Zeit noch Gegenstand von Forschung und Wissenschaft.