



Technische Universität München
Lehrstuhl für Computergestützte Modellierung und Simulation
Ingenieur fakultät Bau Geo Umwelt



Rheinisch-Westfälische Technische Hochschule Aachen
CAAD – Design Computation
Fakultät für Architektur

FE 01.0195/2015/AGB:

Linked Data

Analyse von Einsatzmöglichkeiten von verbundenen Informationen (Linked Data) und Ontologien und damit befassten Technologien (Semantic Web) im Bereich des Straßenwesens

Abschlussbericht

Jakob Beetz
Julian Amann
André Borrmann

Beauftragt und gefördert durch die
Bundesanstalt für Straßenwesen (BASt)

München / Eindhoven, 28.02.2018

Kontakt

Technische Universität München
Ingenieurfacultät Bau Geo Umwelt
Lehrstuhl für Computergestützte Modellierung und Simulation

Arcisstraße 21
80333 München
<https://www.cms.bgu.tum.de>

Prof. Dr.-Ing. André Borrmann

Telefon: +49 89 289-23047

Telefax: +49 89 289-25051

E-Mail: andre.borrmann@tum.de

Julian Amann, MSc

Telefon: +49 89 289-25062

Telefax: +49 89 289-25051

E-Mail: julian.amann@tum.de

RWTH Aachen

Fakultät für Architektur

CAAD – Design Computation

Schinkelstraße 1

52062 Aachen

<https://caad.arch.rwth-aachen.de/>

Prof. Dr.-Ing. Jakob Beetz

Telefon: +49 241 80 95235

Telefax: +49 241 80 92618

E-Mail: beetz@caad.arch.rwth-aachen.de

Inhalt

1	Einleitung	4
1.1	Zielsetzung	4
1.2	Konzept	4
2	Vorgehen	6
2.1	AP 1: Bewertung der Arbeiten zu Linked Data in der Geoinformatik	6
2.1.1	Datenformate in der Geoinformatik	6
2.1.2	Datenformate des Straßenwesens	8
2.1.3	Linked Data Ansätze in der Geoinformatik	15
2.2	AP 2: Analyse von OKSTRA und CB-NL	16
2.2.1	Analyse des OKSTRA-Datenmodells	16
2.2.2	Analyse von CB-NL/OTL	32
2.2.3	Vergleich der Anwendungsgebiete von OKSTRA und CB-NL	41
2.3	AP 3: Die Grundsätze der Modellierung für eine Übertragung des OKSTRA in OWL	43
2.3.1	Stand der Technik	44
2.3.2	AAA, OWA und CWA	44
2.3.3	Schemadaten (UML/XMI/EXPRESS -> OWL)	46
2.3.4	Instanzen	55
2.3.5	Auswirkungen hinsichtlich der Handhabbarkeit, rechnerischen Komplexität und Beweisbarkeit	57
2.4	AP 4: Übersetzung des OKSTRA in OWL	59
2.5	AP 5: Vergleich und Harmonisierung der Definition der Objekte und ihrer Attribute mit Anwendung von Linked Data	59
2.5.1	Mappings	59
2.5.2	Strukturähnlichkeiten	60
2.5.3	Sprachähnlichkeiten	61
2.6	AP 6: Aufbereitung und Anreicherung von RWS Datenbeständen mit OKSTRA Inhalten	63
2.6.1	Fallbeispiel A4-A76/E314	64
2.6.2	Anwendungsfall OKSTRA-CityGML Aachen	67
2.7	AP 7: Erarbeitung weiterführender Modellverwendungen	70
2.7.1	Verknüpfung von OKSTRA-Entwurfsobjekten und OKSTRA-Bestandsobjekten	70
2.8	AP 8: Prototypische Implementierung einer gemeinsamen Datenmodellierung und Repräsentation von OKSTRA-Objekten	74
3	Ergebnisse	78
	Literatur	82

1 Einleitung

1.1 Zielsetzung

In diesem Projekt wurden methodische Ansätze und Prototypen zur Verknüpfung des Objektkatalog für das Straßen- und Verkehrswesen (OKSTRA¹) mit anderen Informationsmodellen auf Basis der Technologien „Linked Data“ und „Semantic Web“ erarbeitet und anhand beispielhafter Datensätze auf ihre Praxistauglichkeit hin überprüft.

In diesem Zusammenhang wurde die Einbeziehung und Kontextualisierung ähnlicher und ergänzender Informationsmodelle im europäischen und internationalen Raum in verschiedenen Anwendungsfällen untersucht sowie das Entwicklungspotential wechselseitiger Anbindungen kritisch ausgelotet.

Im Zuge des Projekts wurde dabei überprüft, inwiefern der OKSTRA eine mögliche Rolle als zukünftiges Grundmodell eines gesamteuropäischen Modells einnehmen kann. Zunächst auf bi-nationale (DE/NL) Beispiele beschränkt, wurde im Projekt eine technische Machbarkeitsstudie grenzüberschreitender Analysen von Straßeninformationen erarbeitet, die die Möglichkeiten und Einschränkungen der Verwendung von Linked-Data-Technologien demonstriert. Erkenntnisse und Ergebnisse wurden darüber hinaus auch auf ihre mögliche Bedeutung für den föderalen bzw. länderübergreifenden Einsatz begutachtet, die eine Vernetzung der verschiedenen nationalen Systeme zulässt.

1.2 Konzept

Im Kern des Untersuchungskonzepts stand die Überführung des bislang in UML modellierten und auf XML-Schema abgebildeten Datenmodells OKSTRA in die Web Ontology Language (OWL). Für die Überführung gibt es unterschiedliche Herangehensweisen, deren jeweiligen Vor- und Nachteile dokumentiert wurden, bevor eine Abbildung ausgewählt und umgesetzt wurde. Auf der Basis des resultierenden Datenmodells okstraOWL wurden anschließend Verfahren des Linked Data angewandt, um beispielhaft die niederländische Objektbibliothek CB-NL mit dem OKSTRA zu verbinden. Diese Verknüpfung wurde auf Instanzebene mithilfe von realen Datensätzen erprobt und der Gesamtansatz anhand der Formulier- und Verarbeitbarkeit von modellübergreifenden Anfragen evaluiert.

Das Gesamtkonzept ist in Abbildung 1 dargestellt.

Im Einzelnen ergaben sich folgende Arbeitspakete bzw. -schritte:

1. Erfassung des Stands der Wissenschaft zu Linked Data, mit Fokus auf die Bereiche Geoinformatik und Building Information Modeling (BIM)
2. Analyse der Architektur der Ontologie, der Semantik und des Inhalts von OKSTRA und CB-NL
3. Erarbeitung von Grundsätzen zur Modellierung für eine Übertragung des OKSTRA in OWL
4. Übersetzung des OKSTRA in OWL: Generierung von okstraOWL
5. Vergleich und Harmonisierung der Definition der Objekte in okstraOWL und CB-NL
6. Aufbereitung und Anreicherung von Datenbeständen der niederländischen Straßenbaubehörde Rijkswaterstaat (RWS) mit OKSTRA-Inhalten
7. Erarbeitung weiterführender Modellverwendungen
8. Prototypische Implementierung einer gemeinsamen Datenmodellierung und Repräsentation von okstraOWL-Objekten

¹ Auf die Verwendung des Markenregister-Zeichens OKSTRA® wird aus Gründen der Lesbarkeit im gesamten Dokument verzichtet.

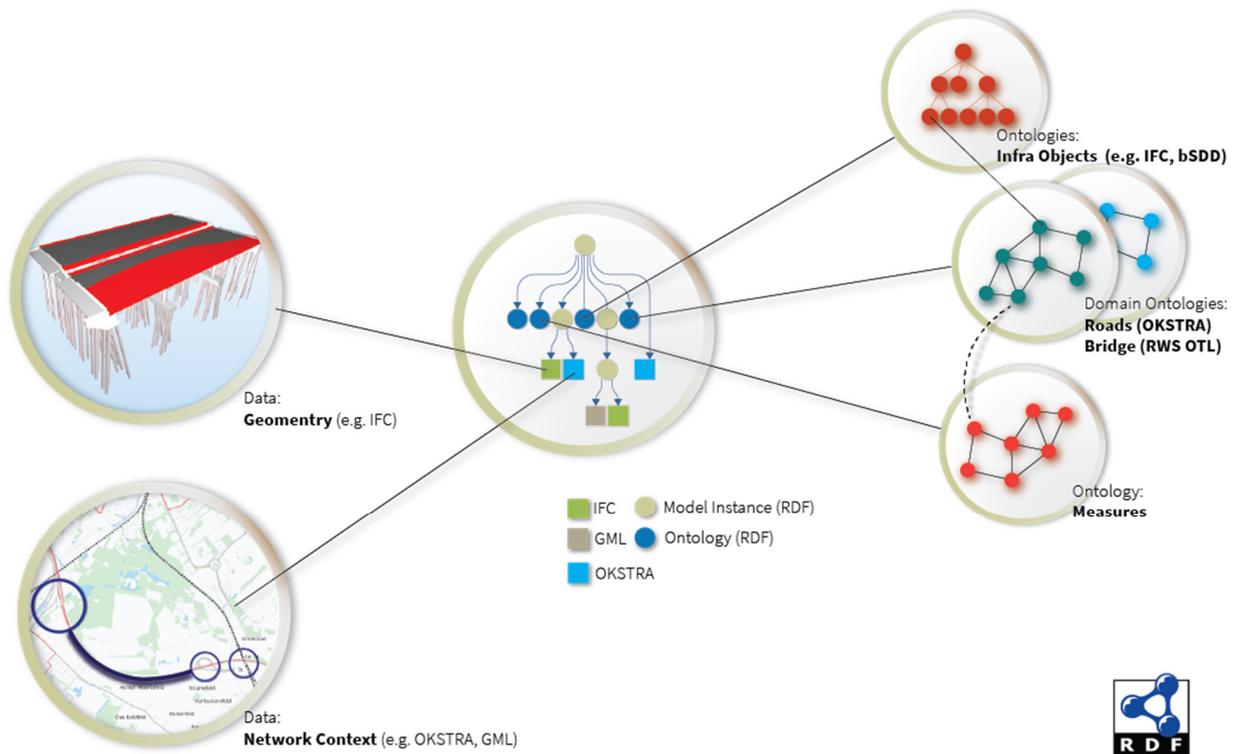


Abbildung 1: Gesamtkonzept zur möglichen Nutzung von Linked-Data-Technologien im Straßenwesen

2 Vorgehen

2.1 AP 1: Bewertung der Arbeiten zu Linked Data in der Geoinformatik

In diesem Arbeitspaket wurde zunächst eine detaillierte Übersicht über die Gesamtsituation der Datenverarbeitung in Straßenwesen und Infrastruktur ausgearbeitet. Diese Übersicht geht näher auf einzelne heute verwendete Modelle in der Geoinformatik (GML, CityGML, InfraGML), des Bereichs der Infrastruktur und des Straßenwesens (OKSTRA, LandXML, IfcAlignment, IfcRoad) sowie auf die existierenden Modelle im Hochbau (IFC, ifcOWL) ein, die für weitere Vernetzungen mittels Linked Data Prinzipien infrage kommen oder sich bereits in der Entwicklung befinden.

Neben dem vom OGC standardisierten und in mehreren Software-Plattformen implementierten GeoSPARQL Standard mit seiner effektiven geometrischen Beschreibung als Well Known Text (WKT²) und seiner 2D räumlichen Algebra werden dabei auch neuere Entwicklungen wie etwa der LinkedGeoData Initiative betrachtet. Dabei wird speziell auf das Potential hinsichtlich einer Verwendbarkeit und Erweiterbarkeit im Kontext von OKSTRA eingegangen.

2.1.1 Datenformate in der Geoinformatik

Datenformate der OGC

Im Bereich der Geoinformatik ist das Open GeoSpatial Consortium (OGC) das maßgebliche Standardisierungsgremium für die Definition und Weiterentwicklung von Datenaustauschstandards. Das OGC umfasst alle wesentlichen Softwarehersteller der Geoinformatik sowie große Anwendervereinigungen. Häufig werden die dort entwickelten Standards später in ISO-, CEN- und DIN-Normen überführt. Zu den wichtigsten OGC-Standards gehören:

- Geography Markup Language (GML)
- Keyhole Markup Language (KML)
- OpenGIS Location Service (OLS)
- Web Feature Service (WFS)

Im hier betrachteten Anwendungsfall hat der Standard GML die größte Bedeutung. Bei GML handelt es sich um ein in XML-Schema definiertes Datenmodell zur Beschreibung geographischer Daten und raumbezogener Objekte (Features). GML erlaubt die Übermittlung von Objekten mit Attributen, Relationen und Geometrien. Die aktuelle Version trägt die Versionsnummer 3.2.1. Die OGC-Spezifikation für GML 3.2 ist identisch mit DIN EN ISO 19136. Der OKSTRA-UML-Standard macht zur Definitionen der Lage und der Geometrie von Objekten mit Raumbezug Gebrauch vom Standard ISO 19107. Im daraus abgeleiteten XML-Schema kommen unmittelbar die Typen des GML-Standards zum Einsatz.

Auf der Basis von GML wurden bzw. werden von OGC zwei weiter spezialisierte Schemata definiert:

- CityGML
- InfraGML

CityGML ist ein auf GML basierender Standard zur Beschreibung von 3D-Stadtmodellen (Kolbe et al. 2005). CityGML beinhaltet neben den notwendigen 3D-Geometriebeschreibungen vor allem auch semantische Informationen für die Verwendbarkeit der entstehenden Modelle für verschiedene Einsatzzwecke. Die dreidimensionale Modellierung von ober- oder unterirdischer Infrastruktur wird ansatzweise in den Application Domain Extensions (ADE) Bridge (SIG 3D 2009a) und Subsurface Structure berücksichtigt. Diese dienen jedoch vornehmlich der Beschreibung bereits existierender Bauwerke. Das gleiche gilt für die seit Version 2.0 integrierten Möglichkeiten zur Beschreibung von Tunneln.

² <http://docs.opengeospatial.org/is/12-063r5/12-063r5.html>

Seit 2014 befindet sich der Standard InfraGML in der internationalen Entwicklung. Hierzu wurde die Land and Infrastructure Standards Working Group (SWG) gegründet³. Ziel der Entwicklung ist die Bereitstellung eines Datenmodells, das die Beschreibung und Übertragung von Anlagen der baulichen Infrastruktur auf Basis von GML erlaubt. Zum Definitionsumfang gehören dabei Straßen ebenso wie Schienentrassen, wobei unterschiedliche Konformitäts- bzw. Anforderungsklassen definiert werden (Abbildung 2).

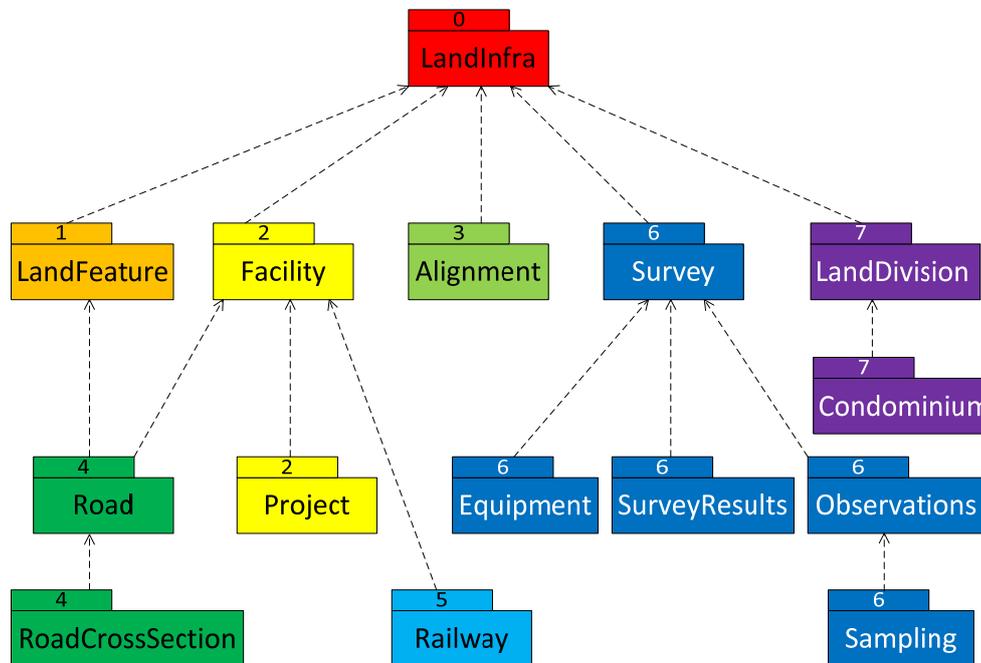


Abbildung 2: LandInfra Requirements Classes einschließlich der Abhängigkeiten zwischen den verschiedenen InfraGML-Teilen (Quelle: OGC)

Eine wesentliche Komponente von InfraGML ist die Definition von Trassierungen mithilfe ingenieurtechnischer Trassierungselemente im Grundriss (Gerade, Kreisbogen und Übergangskurve) und im Längsprofil (Gerade, Ausrundungskurve). Zudem kann eine lineare Positionierung entlang der entstehenden Achse vorgenommen und Querschnitte definiert werden. Für den Standard wurde zunächst in enger Kooperation mit buildingSMART ein konzeptionelles Datenmodell⁴ erarbeitet. Seit August 2017 liegt InfraGML als finaler Standard vor⁵. OGC will mit der Einführung von InfraGML das weitverbreitete Format LandXML ablösen, welches nicht auf GML basiert und einen proprietären Standard darstellt, d.h. nicht von einer Standardisierungsorganisation definiert wird.

Die Abfragesprache SPARQL gehört neben RDF und OWL zu den Bausteinen des Semantic Web. Mittels SPARQL können RDF-Datensätze abgefragt werden. Der OGC-Standard GeoSPARQL⁶ dient der Abbildung von GML-Daten als RDF/OWL-Datensätze sowie der Möglichkeit zur Abfrage der so abgelegten Informationen mithilfe der Abfragesprache SPARQL. Er definiert hierzu eine Ontologie für geographische Informationen auf der Basis des General Feature Model, Simple Features und Feature Geometry sowie eine Menge von SPARQL-Filterfunktionen. Letzteres schließt u.a. die Nutzung von topologischen Beziehungen (Egenhofer-Modell und RC8-Modell) als Filter im Rahmen einer Abfrage ein. Heraushebenswert ist auch die effektive Beschreibung geometrischer Informationen als Well Known Text (WKT). Dies verringert sehr effektiv den Overhead, der bei unmittelbarer Anwendung von RDF auf die sehr umfangreichen geometrischen Informationen entstehen würde. Voraussetzung für den Einsatz von GEOSPARQL ist die Nutzung eines sog. Endpoint, der derartige Anfragen unterstützt und verarbeiten kann, insbesondere in Hinblick auf die geographischen Filterfunktionen.

³ <http://www.opengeospatial.org/projects/groups/landinfraswg>

⁴ <http://docs.opengeospatial.org/is/15-111r1/15-111r1.html>

⁵ <http://www.opengeospatial.org/standards/infragml>

⁶ <http://www.opengeospatial.org/standards/geosparql>

2.1.2 Datenformate des Straßenwesens

2.1.2.1 OKSTRA

Beim Objektkatalog für das Straßen- und Verkehrswesen (OKSTRA) handelt es sich um ein in Deutschland standardisiertes Datenmodell für die Datenaustauschprozesse des Straßenwesens, der alle Phasen des Lebenszyklus umfasst, einschließlich des Entwurfs, des Betriebs und der Erhaltung von Straßen sowie der Erfassung von Verkehrsdaten.

Der OKSTRA-Standard wurde vom Bundesministerium für Verkehr, Bau und Stadtentwicklung im Jahr 2000 als bundesweiter Standard eingeführt. Die Einführung des Standards wurde mit dem „Allgemeinen Rundschreiben Straßenbau 12/2000“ des Bundesverkehrsministeriums für den Bereich der Bundesfernstraßen umgesetzt. 2011 wurde die Version 1.015 verabschiedet, die derzeit noch sehr weit verbreitet ist. Die letzte veröffentlichte Version 2.017 erschien im April 2016.

Die Hauptmotivation zur Entwicklung des OKSTRA bestand darin, die IT-seitige Unterstützung der für die Verwaltung und für den Bau von Bundesfernstraßen notwendigen Prozesse zu verbessern. Um das Straßennetz optimal unterhalten, ändern und ausbauen zu können, werden in verschiedenen Prozessen Daten, wie z. B. Verkehrsdaten, Zustandsdaten, Umweltdaten, Unfalldaten oder Wirtschaftlichkeitsdaten, erhoben und analysiert. Basierend auf diesen Daten und Analysen werden gegebenenfalls Maßnahmen, wie Erhaltungsmaßnahmen, Neubaumaßnahmen oder administrative Maßnahmen, umgesetzt.

Die Koordination des OKSTRA-Projektes erfolgt durch eine Fachgruppe des Bund-/Länder-Fachausschusses "IT-Koordinierung", die sogenannte FG OKSTRA. Die Leitung liegt bei der Bundesanstalt für Straßenwesen (BASt). Die Pflege des OKSTRA-Standards wird durch die Firma interactive instruments GmbH durchgeführt.

Für die Definition des OKSTRA-Datenmodells wurde zunächst (ebenso wie bei STEP und IFC) die Datenmodellierungssprache EXPRESS eingesetzt, die jedoch ab der Version 2.0 vollständig von UML als primäre Datenmodellierungssprache abgelöst wurde. Aus UML wird anschließend XML-Schema⁷ abgeleitet. Der OKSTRA besteht aus verschiedenen Fachschalen, u.a. zum Entwurf, zur Mengenermittlung, zur Landschaftsplanung, zum Verkehr. Jeder dieser Fachschalen ist in einem eigenen XML-Schema definiert, die modular aufeinander aufbauen.

Als grafische Modellierungssprache wurden ursprünglich NIAM-Diagramme verwendet. Inzwischen wird die Unified Modeling Language (UML) eingesetzt, die eine automatisierte Abbildung in verschiedene Datenmodellierungssprachen erlaubt, darunter XML-Schema. Das UML-Datenmodell steht im Neutralformat XML Metadata Interchange (XMI) zur Verfügung⁸. Abbildung 3 gibt einen Überblick über die verschiedenen OKSTRA-Versionen und die verwendeten Modellierungssprachen der entsprechenden Version.

Version	XML	UML	EXPRESS	NIAM	SQL
1.000			X	X	X
1.001			X	X	X
1.002			X	X	X
1.003			X	X	X
1.004			X	X	X
1.005			X	X	X
1.006			X	X	X

⁷ XML-Schema ist eine Spezifikation des World Wide Web Consortium (W3C): <https://www.w3.org/TR/xmlschema-1/>

⁸ XMI ist ein Standard der Object Management Group (OMG).

1.007	X		X	X	X
1.008	X		X	X	X
1.009	X		X	X	X
1.010	X		X	X	X
1.011	X		X	X	X
1.012	X		X	X	
1.013	X		X	X	
1.014	X		X	X	
1.015	X		X	X	
2.015 ⁹	X	X			
2.016	X	X			
2.017	X	X			

Abbildung 3: Historie der OKSTRA-Datenmodellierung

Seit der Version 2.015 setzt der OKSTRA zur Beschreibung geometrischer (bzw. geographischer) Informationen auf dem oben beschriebenen GML-Standard (Version 3.2.1) auf. Damit ist es grundsätzlich denkbar, die Anfragesprache GeoSPARQL zur Abfrage von OKSTRA-Daten einzusetzen. Zum vollständigen Erschließen von OKSTRA-Datensätzen, einschließlich der sehr wichtigen semantischen Informationen, ist es jedoch unumgänglich, das OKSTRA-Datenmodell zuvor in OWL/RDF zu konvertieren. Dies wird im vorliegenden Bericht in Abschnitt 2.3 dokumentiert.

Ein fundamentales Problem von OKSTRA, das einer EU- bzw. weltweiten Verwendung entgegensteht und gegen eine Veröffentlichung von OKSTRA als eine buildingSMART Publicly Available Specification (PAS) spricht, ist die deutschsprachige Dokumentation bzw. das deutschsprachige Datenmodell. In diesem Zusammenhang wurde in einem Forschungsprojekt der BASt ein erster Teil des Standards in die englische Sprache samt Beispieldaten übersetzt. Die Übersetzung wurde auf OKSTRA-Webseite veröffentlicht¹⁰.

2.1.2.2 LandXML

LandXML ist ein XML-Dateiformat für den Austausch und die Planung von Verkehrswegen für das Bauingenieur- und Vermessungswesen. Es handelt sich um ein offenes Datenformat, das allerdings nicht von einem offiziellen Standardisierungsgremium definiert und gepflegt wird. Man spricht daher auch von einem Quasi- oder Industriestandard.

Das Format folgt grundsätzlich dabei dem klassischen Ansatz der Trassenplanung basierend auf 2D-Ansichten eines Lageplans und eines Höhenplans. Zusätzlich werden Querschnitte der Fahrbahn definiert. Die 2D-Informationen der verschiedenen Ansichten können kombiniert werden, um daraus ein 3D-Modell des Trassenverlaufs bzw. der Straße zu generieren (Abbildung 4).

Im Juli 2002 wurde LandXML-1.0 als Industriestandard von LandXML.org ratifiziert. Das LandXML Schema basiert auf den W3C XML Schema 1.0, das am 2. Mai 2001 offiziell von der W3C freigegeben wurde. Gleichzeitig wurde mit LandXML-1.0 das LandXML Software Developer Kit 1.0 veröffentlicht. Im Juli 2006

⁹ Die Version 2.015 ist keine offizielle OKSTRA-Version, sondern sollte als "Ansichtsversion" nur einen Eindruck davon vermitteln, wie die Version 1.015 in UML aussieht.

¹⁰ http://www.okstra.de/forschung/okstra_internationalization_EN.html

wurde LandXML-1.1 ratifiziert. Ebenfalls wurde das LandXML Software Developer Kit 1.1 veröffentlicht. Ein wichtiges Update erhielt LandXML im Juli 2008 durch die Vorstellung des Working Draft der Version 1.2.

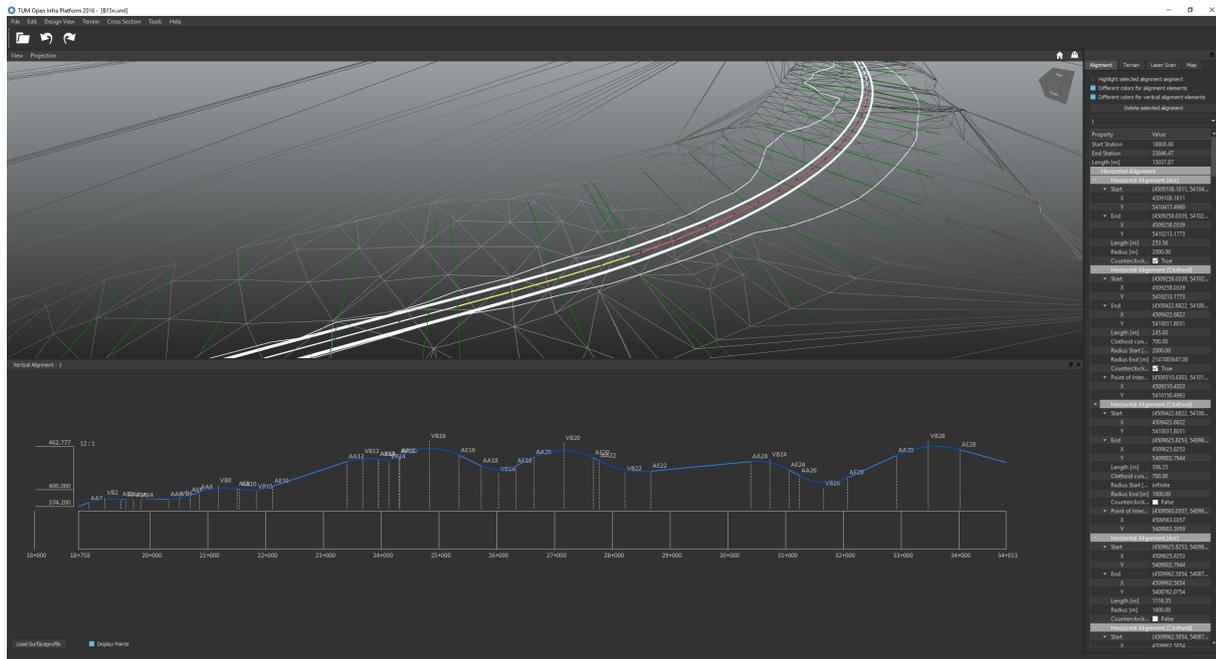


Abbildung 4: LandXML-Trasse in der TUM Open Infra Platform

Die Arbeiten an LandXML wurden durch landxml.org von November 2012 bis Ende 2015 pausiert. 2013 entschloss sich das Open Geospatial Consortium (OGC), den Standard für die mögliche Aufnahme als OGC-Standard zu untersuchen. Nach einer Analyse von LandXML durch die OGC wurden zahlreiche formale Mängel an LandXML festgestellt und man entschloss sich, im Rahmen der OGC einen eigenen, von der Funktionalität vergleichbaren Standard zu entwickeln. Im April 2016 wurde auf der LandXML-Webseite die Version LandXML 2.0 vorgestellt. Jedoch scheint die Version 1.2 bzw. 1.1 noch häufiger in Gebrauch zu sein und in der Praxis von größerer Relevanz zu sein. Die Unterstützung für LandXML 2.0 ist noch sehr unausgereift.

Eine der größten Schwächen von LandXML ist die ungenaue Dokumentation. LandXML besitzt kein zugrundeliegendes konzeptionelles Modell und lässt daher viel Spielraum für „Implementation defined behaviour“, was einen reibungslosen Datenaustauschprozess erschwert bzw. unmöglich macht. Jeder Hersteller verwendet beispielsweise eigene Beschreibungen für Querschnitte. Darüber hinaus ist es nicht möglich parametrisierte Trassen, die beispielsweise eine einfache Anpassung der Querschnitte möglich machen würden, zu spezifizieren. Gleichzeitig muss auch die Herstellerneutralität gewährleistet werden.

Dennoch hat sich LandXML in der Version 1.2 als De-facto-Standard für den Austausch von Trassierungsinformationen im internationalen Bereich durchgesetzt und ist weit verbreitet. LandXML hat jedoch beim Austausch von z. B. Querprofildaten erhebliche Nachteile zu verzeichnen. So wird lediglich das Ergebnis einer Querprofilkonstruktion an bestimmten Stationierungspunkten übergeben. Die Regeln und Konstruktionsprozesse zur Erstellung des Straßenkörpers sind nicht Teil des Datenformats, was eine Weiterbearbeitung in einer anderen Softwareapplikation erschwert.

Mit der Version 1.1 wurde gegenüber der Version 1.0 die Abbildung von Querprofildaten bedeutend verbessert. Das Konzept wird im Folgenden vorgestellt. Zur Abbildung des Straßenkörpers wurden sogenannte *Design Cross Sections* eingeführt.

```
<CrossSects>
  <CrossSect sta="18808.601999999999" name="18+808.602">
    <CrossSectSurf name="P1-dnetz01" desc="">
      <PntList2D>-60. 0. -0.000000000082 0. 59.999999999835 0.</PntList2D>
    </CrossSectSurf>
    <DesignCrossSectSurf name="1. Deckschicht" side="right" closedArea="true">
```

```
area="0.1">
  <CrossSectPnt>0. 0.</CrossSectPnt>
  <CrossSectPnt code="Fahrspurrand_Außen">4. -0.08</CrossSectPnt>
  <CrossSectPnt code="Fahrspurrand_Außen_Deckschicht">4. -0.105</CrossSectPnt>
  <CrossSectPnt>0. -0.025</CrossSectPnt>
  <CrossSectPnt>0. 0.</CrossSectPnt>
</DesignCrossSectSurf>
```

Abbildung 5: Instanz eines Querprofils in einer LandXML-Datei

In Abbildung 5 ist ein in LandXML definiertes Querprofil exemplarisch dargestellt. Das bereits in Version 1.0 enthaltene `CrossSectSurf`-Element dient zur Abbildung des natürlichen Geländes. Mit Hilfe der `DesignCrossSectSurf` wird der Straßenaufbau beschrieben. Mehrere dieser Elemente werden in einem `CrossSect`-Element zusammengefasst, welches eindeutig einer Stationierung zugeordnet ist. Die `DesignCrossSectSurf`-Elemente können sowohl geschlossene Volumen als auch Flächen beschreiben. Jeder Punkt eines `DesignCrossSectSurf`-Elements kann auf zwei Arten eindeutig gespeichert werden. Zum einen über den Abstand und Höhendifferenz zur Achse zum anderen über Länge und Neigung des Abschnitts bezogen auf die Achse oder den vorherigen Punkt. In Abbildung 6 sind gerenderte `CrossSectSurf`- und `DesignCrossSectSurf`-Elemente als Teile von Querprofilen zu sehen.

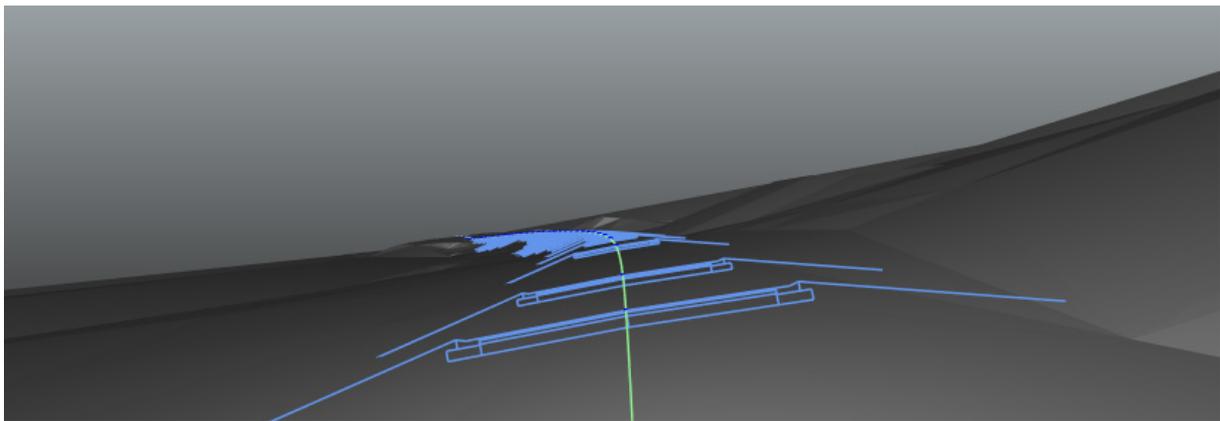


Abbildung 6: LandXML Cross Section mit Hilfe der TUM Open Infra Plattform gerendert

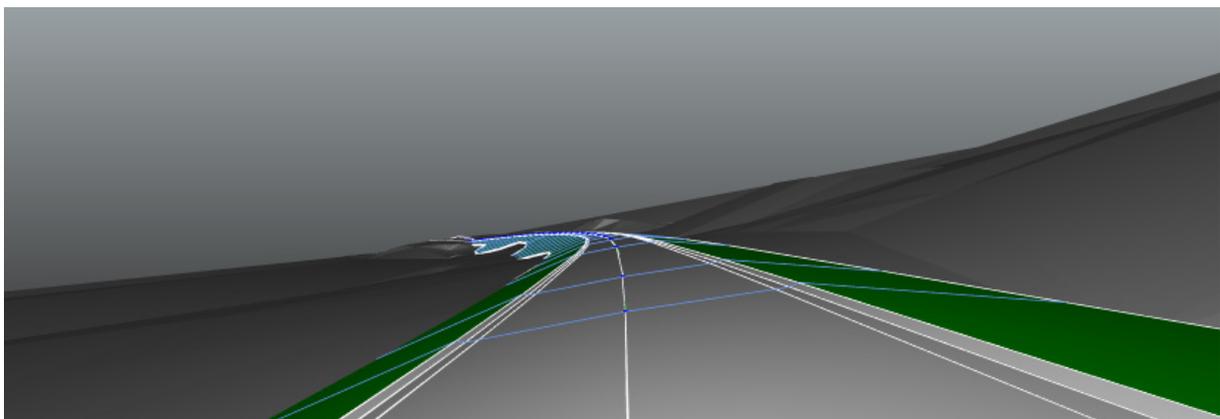


Abbildung 7: LandXML Straßenkörper mit Hilfe der TUM Open Infra Plattform gerendert

Mit Hilfe der *Design Cross Section* Elemente kann das LandXML-Format das Modell einer Straße implizit¹¹ dreidimensional beschreiben, alle Daten zur Absteckung im Feld bereitstellen, als Grundlage für Flächen-

¹¹ Die 3D-geometrie ergibt sich aus der Auswertung von Lageplan, Längsprofil und Querprofilen.

und Massenermittlungen dienen und ggf. auf nicht-lineare Weise einen Straßenkörper zwischen den diskreten Querprofilen definieren.

Zur Beschreibung des Straßenkörpers zwischen den diskreten Querprofilen sieht das LandXML-Format Referenzattribute vor. So kann jeder `CrossSectPnt` mit Hilfe des `alignRef`-Attributs zusätzliche Geometrie, nämlich ein `Alignment`-, `Parcel`- oder `PlanFeature`-Element, referenzieren. Weiter kann zusätzlich die Stationierung des `CrossSectPnt` entlang der referenzierten Geometrie durch das Attribut `alignRefStation` angegeben werden. In Abbildung 7 ist das Rendering eines Straßenkörpers einer LandXML Datei zu sehen.

Größter Nachteil des LandXML-Formats ist die Tatsache, dass zwar wie beschrieben die Geometrie eines Straßenkörpers komplett abgebildet werden kann, die Regeln und Prozesse, wie diese Geometrie zustande kommt, allerdings nicht weitergegeben werden. Dies liegt ausdrücklich nicht im Scope von LandXML.

Das Datenmodell von LandXML ist im Lauf der Zeit gewachsen. Abbildung 8 gibt einen Überblick über den Anstieg bei der Anzahl der Klassen, Aufzählungstypen und Typen im LandXML-Quasistandard. Darüber hinaus zeigt Abbildung 9 den Anstieg bei der Anzahl von Attributen im LandXML-Datenmodell.

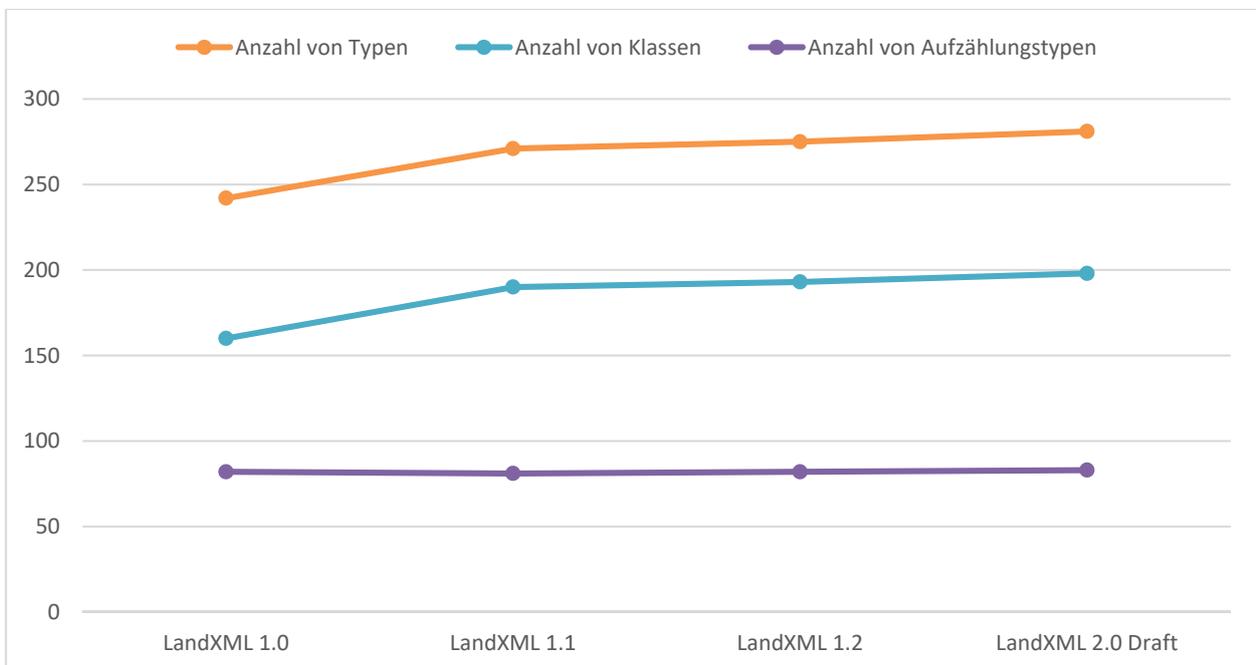


Abbildung 8: Anstieg bei der Anzahl der Klassen, Aufzählungstypen und der Anzahl der Attribute im LandXML Quasistandard

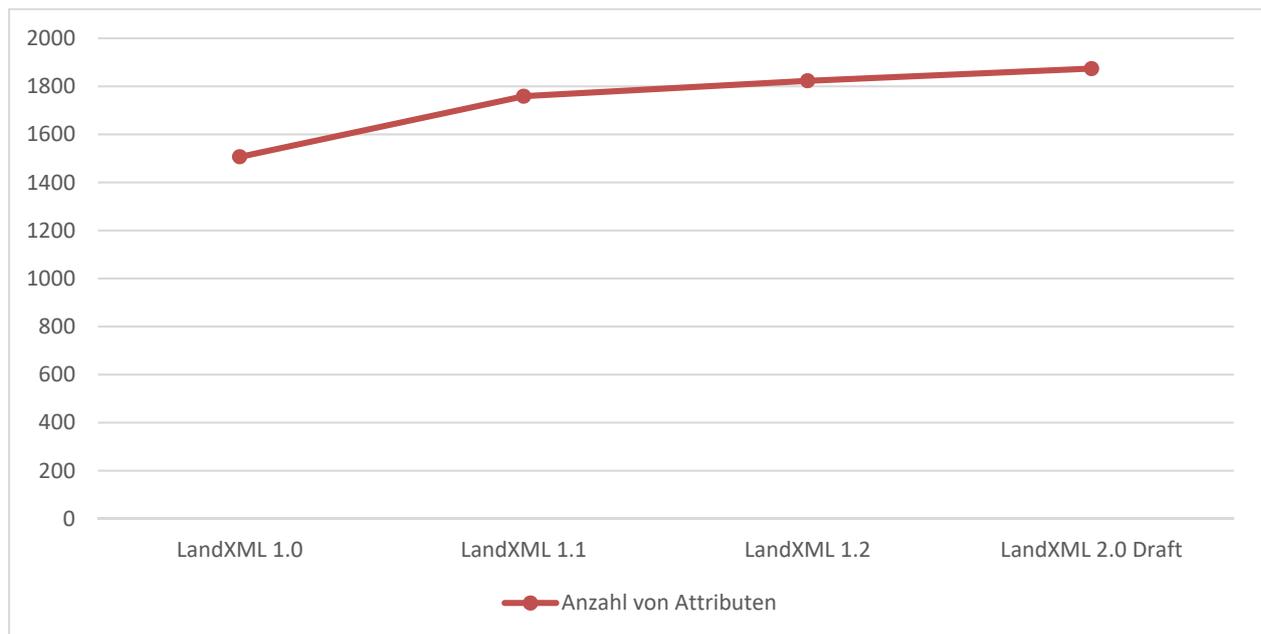


Abbildung 9: Anzahl von Attributen in den verschiedenen Versionen von LandXML

2.1.2.3 buildingSMART Industry Foundation Classes

buildingSMART ist eine internationale non-for-profit Organisation, die sich zum Ziel gesetzt hat, offene Datenaustauschstandards für das Bauwesen zu schaffen. Im Fokus steht dabei der gesamte Lebenszyklus von den frühen Entwurfsphasen eines Bauvorhabens über die Bauausführung bis hin zum Betrieb bzw. der Instandhaltung des Bauwerks.

Der wichtigste von buildingSMART entwickelte Standard ist das Datenmodell Industry Foundation Classes (IFC). Er erlaubt es, Bauwerke und damit assoziierte Informationen in sehr umfassender und detaillierter Weise nach dem objektorientierten Paradigma zu beschreiben. Das Datenmodell trennt dabei strikt zwischen semantischer und geometrischer Beschreibung. Die geometrische Beschreibung umfasst eine große Menge verschiedener Techniken, darunter sowohl explizite Verfahren (Boundary Representation - trianguliert, Vollflächen, NURBS) wie auch implizite (Extrusion, Sweeping, Constructive Solid Geometry).

Die letzte offiziell verabschiedete Version der IFC ist die Version 4.1¹². Sie wird momentan nach und nach von führenden Softwareherstellern implementiert. Das IFC-Datenmodell wird zurzeit noch primär mit der Datenmodellierungssprache EXPRESS definiert, es existieren aber Mappings auf XML-Schema (genannt ifcXML¹³) und auf OWL (genannt ifcOWL¹⁴). Es steht zu erwarten, dass EXPRESS mittelfristig durch XML-Schema als primäres Datenmodellierungsmittel abgelöst wird. Die Verfügbarkeit der OWL-Abbildung des Datenmodells erlaubt schon jetzt die direkte Anwendung von Linked-Data-Technologien (Beetz et al., 2009).

Bislang hat sich die Entwicklung des IFC-Format nahezu ausschließlich auf den Hochbau fokussiert. Da international jedoch ein großes Interesse an der Nutzung von IFC für den Infrastrukturbereich besteht, wurde im Jahr 2010 der Infra Room ins Leben gerufen, eine Abteilung von buildingSMART, die das Ziel hat, IFC-Erweiterungen für die gebaute Infrastruktur (Straßenbau, Schienenbau, Wasserwegebau) vorzunehmen.

Als erstes Projekt des Infra Room wurde IfcAlignment umgesetzt, in dessen Ergebnis die im Frühjahr 2018 verabschiedete Version 4.1 der IFC-Datenmodells entstand, das alle wesentlichen Elemente für die Be-

¹² <http://www.buildingsmart-tech.org/ifc/IFC4x1/final/html/>

¹³ <http://www.buildingsmart-tech.org/specifications/ifcxml-releases>

¹⁴ <http://openbimstandards.org/standards/ifcowl/>

schreibung von Trassierungen beinhaltet. Auch hier wird der im Ingenieurwesen etablierte Ansatz der getrennten Beschreibung von Achse (IfcHorizontalAlignment) und Gradiente (IfcVerticalAlignment) umgesetzt. Es werden in beiden Fällen originäre Kurven für die Beschreibung der Trassierung eingesetzt, wie beispielsweise Klothoiden und Bloss-Bögen im Fall der Übergangskurven bzw. Parabeln oder Kreisbögen für die Ausrundung von Kappen bzw. Wannsen. Das Trassierungsmodell von IFC kann damit als gleichwertig bzw. gleich mächtig wie das Trassierungsmodelle von OKSTRA oder LandXML eingeschätzt werden.

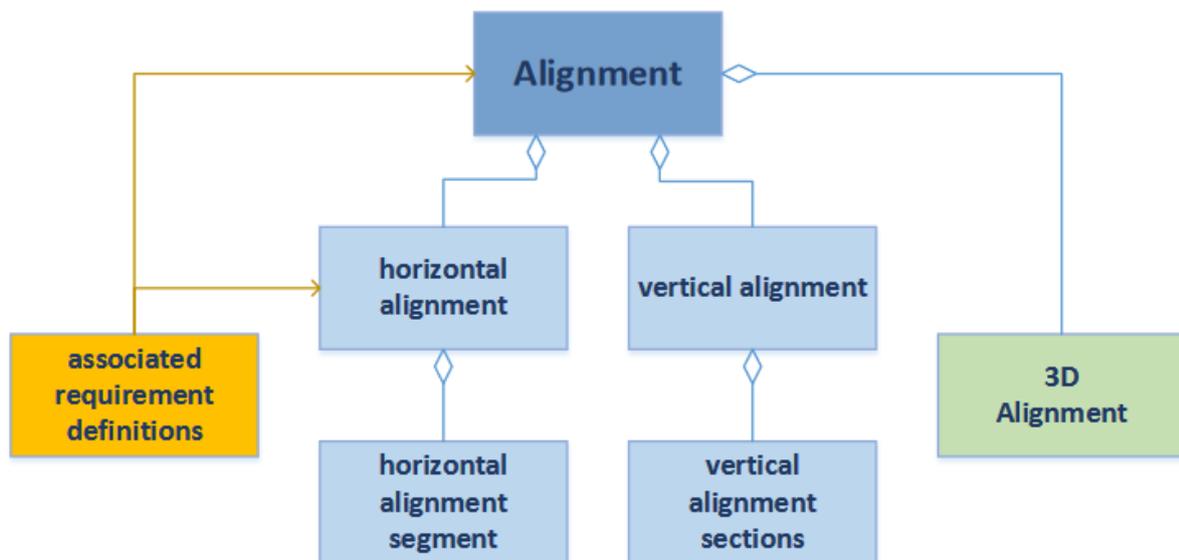


Abbildung 10: IfcAlignment basiert auf der getrennten Beschreibung von Achse (IfcHorizontalAlignment) und Gradiente (IfcVerticalAlignment)

Aufbauend auf den Ergebnissen des Alignment-Projekts werden nun die Projekte IfcRoad, IfcRailway und IfcBridge durchgeführt. Im Kontext des vorliegenden Linked-Data-Projekts ist vor allem das Projekt IfcRoad von Bedeutung. Hier wurden vom koreanischen Forschungsinstitut KICT zwischen 2011 und 2016 umfangreiche Vorarbeiten geleistet, die in der Verabschiedung eines nationalen, IFC-basierten Standards gemündet haben. Dieser Standard umfasst neben einer umfangreichen semantischen Beschreibung von Straßen und deren Komponenten auch eine räumliche Hierarchie. Als Geometrirepräsentation werden jedoch fast ausschließlich explizite Verfahren eingesetzt, mit der Konsequenz das mit diesem Modell nur eine eingeschränkte Menge von Anwendungsszenarien umgesetzt werden können (z.B. Mengenermittlung).

Der koreanische Standard wurde in der ins Englische übersetzten Fassung von buildingSMART als bSI SPEC übernommen. Dies bedeutet lediglich, dass er als Grundlage für die Entwicklung des internationalen IfcRoad-Standards herangezogen werden kann, und nicht, dass er direkt als solcher übernommen wird. In der Tat weist der koreanische Entwurf eine Reihe von Unzulänglichkeiten auf.

Das internationale IfcRoad-Projekt wurde im Januar 2018 gestartet und hat eine Laufzeit von zwei Jahren. Im Ergebnis werden Erweiterungen des IFC-Datenmodells zur umfassenden geometrisch-semantischen Beschreibung von Straßen entstehen.

2.1.3 Linked Data Ansätze in der Geoinformatik

Die Initiative LinkedGeoData (linkedgeo.org) hat sich zum Ziel gesetzt, geographische Informationen für das Semantic Web zugänglich zu machen. Dazu nutzt es die umfangreichen geographischen Informationen, die im Rahmen des OpenStreetMap-Projekts erfasst und bereitgestellt werden. Diese Daten werden als RDF-Wissensbasis bereitgestellt und mit anderen Wissensbasen wie DBpedia und GeoNames verknüpft. Der gesamte RDF-Datensatz besteht aus 20 Milliarden Tripeln. Die Initiative stellt einen SPARQL-Endpoint zur Verfügung, um die Daten effektiv abzufragen. SPARQL-Anfragen lassen sich formulieren und absenden. Da für die Bereitstellung dieses öffentlichen Endpoints eine Softwarelösung verwendet wird, die die meisten GeoSPARQL Features implementiert, können die geoinformatischen Erweiterungsfilter hier genutzt werden. Neben der hier zum Einsatz kommenden Lösung "Virtuoso" der Firma OpenLink Software werden diese Möglichkeiten von weiteren namenhaften Herstellern wie Oracle unterstützt.

2.2 AP 2: Analyse von OKSTRA und CB-NL

In diesem Arbeitspaket wurden die in AP 1 erarbeiteten Übersichten für den OKSTRA und die CB-NL/OTL vertieft und eingehend hinsichtlich mehrerer Aspekte analysiert. Dazu gehören neben den qualitativen Beschreibungen der jeweiligen Modellierungsprinzipien und vereinheitlichten Modelldiagramme auch qualitative Studien der jeweiligen semantischen Tiefe, wie z.B. die Anzahl und inhaltlichen Abdeckung der Objektklassen, ihrer Attribute und Beziehungen sowie die Tiefe und Verästelung ihrer Objektbäume.

Für den OKSTRA wurden auch seine geometrische Beschreibungsformen etwa von Trassenkurven oder Querprofilen untersucht und in Bezug auf ihre Alleinstellung und Weiterverwendbarkeit in anderen Modellen besprochen. Ein Vergleich und die Herausarbeitung des Potentials gegenseitiger Ergänzung und Erweiterung bilden den Abschluss dieses Arbeitspaketes.

2.2.1 Analyse des OKSTRA-Datenmodells

2.2.1.1 Überblick

Auf der Webseite www.okstra.de ist eine detaillierte Dokumentation des OKSTRA-Standards zu finden. Der Standard erstreckt sich von Fachobjekten zur Entwurfsplanung bis hin zu Objekten zur Erfassung von Unfallstatistiken und Polizeidienststellen. OKSTRA ist damit einer der umfangreichsten Standards dieser Art, die sich aktuell in Verwendung befinden.

Die Basiskonzepte von OKSTRA sind im Dokument N0135 zusammengefasst¹⁵, dessen Inhalte im Folgenden ausschnittshaft wiedergegeben werden sollen. Die folgenden Absätze beinhalten zum Teil wörtlich übernommenen Text.

Das XML-Schema der OKSTRA-Version 2.017 basiert auf einer Vielzahl von anderen Standards bzw. Schemata, die im Folgenden aufgelistet sind:

- ISO 00639 Language Codes
- ISO 03166 Country Codes
- ISO 19103:2005 Schema Language
- ISO 19107:2003 Spatial Schema
- ISO 19108:2006 Temporal Schema
- ISO 19109 Application Schema
- ISO 19111 Referencing by Coordinates
- ISO 19115:2006 Metadata (Corrigendum)
- ISO 19136 GML
- ISO 19139 Metadata – XML Implementation
- ISO 19148 Linear Referencing
- ISO 19156:2011 Observations and Measurements

Die gelisteten Schemata sind Teil des „ISO Harmonized Model“.

Alle Klassen („Objektarten“) erben stets direkt oder mittelbar von der Wurzelklasse OKSTRA_Objekt. Diese Objektart enthält u. a. das Attribut „OKSTRA_ID“ zur optionalen Angabe eines Objektidentifikators in Form einer GUID. Objektarten, die in Assoziationen auch durch die Angabe eines symbolischen Identifikators referenziert werden können, erhalten das Stereotype „FachId“ im UML-Modell.

Im OKSTRA-UML-Modell werden die elementaren Datentypen des Basispakets „ISO 19103:2005 Geographic information - Schema Language“ eingesetzt (Boolean, Integer, Real, CharacterString, Sequence<Bit>

¹⁵ <http://www.okstra.de/docs/n-dokumente/n0135.pdf>

Date, ClockTime). Elementare Datentypen zur Repräsentation von Werten, die in irgendeiner Weise gemessen werden können, werden aus dem Typ „Measure“ bzw. einer seiner Spezialisierungen („Angle“, „Area“, „Length“ etc.) abgeleitet.

In die Kategorie der elementaren Datentypen fallen auch die folgenden Datentypen des Basispakets „ISO 19107:2003 Spatial Schema“ zur Darstellung von Geometrien.

Einfache Geometrie:

- Punktgeometrie: GM_Point,
- Liniengeometrie: GM_Curve,
- Flächengeometrie: GM_Surface,
- Volumengeometrie: GM_Solid.

Mehrfach-Geometrien:

- Punktgeometrien: GM_MultiPoint,
- Liniengeometrien: GM_MultiCurve,
- Flächengeometrien: GM_MultiSurface,
- Volumengeometrien: GM_MultiSolid.

Der GML-Standard (ISO 19136) wird beispielsweise genutzt, um Geometriedaten, wie Raumpunkte, zu beschreiben. Das folgende Beispiel zeigt einen Achshauptpunkt, der Bestandteil eines Achselements ist, im XML-Format:

```
• <okstra:Achshauptpunkt gml:id="Achshauptpunkt_8652">
•   <okstra:Punktgeometrie>
•     <gml:Point gml:id="Point.8670">
•       <gml:pos srsDimension="3">6183.281 5084.336 0.0</gml:pos>
•     </gml:Point>
•   </okstra:Punktgeometrie>
•   <okstra:Ende_von_Achselement   Objektklasse="Achselement"   xlink:href="#Achselement_8637"/>
• </okstra:Achshauptpunkt>
```

Neue elementare Datentypen, die aus den Typen der Basispakete per Vererbung abgeleitet werden, werden mit dem Stereotype „type“ gekennzeichnet

Komplexe Datentypen können ebenso wie die elementaren Datentypen als Typspezifikatoren von Attributen in Objektarten oder anderen komplexen Datentypen verwendet werden. Im Unterschied zu den elementaren Datentypen können sie jedoch mehrere Informationen enthalten. Da ein komplexer Datentyp im Gegensatz zu einer Instanz einer Objektart keine eigenständige Identität besitzt, kann eine von ihm ausgehende Assoziation nur in der Richtung von ihm weg navigiert werden (d.h. er verweist auf eine Objektart, diese aber nicht auf ihn zurück). Komplexe Datentypen sind mit dem Stereotype „DataType“ versehen.

Schlüsseltabellen sind eine Besonderheit der OKSTRA-Modellierung, die eine dynamische Erweiterung des Modells erlauben. Eine Schlüsseltabelle ist ein Datentyp, der die Angabe eines Eintrags aus einem (ggf. bereits im UML-Modell definierten) Wertekatalog ermöglicht. Eine Schlüsseltabelle besitzt zwei oder mehr Attribute vom Datentyp „CharacterString“. Das erste dieser Attribute muss den Rollenname „Kennung“ besitzen und dient zur Abbildung der eindeutigen Kennung des jeweiligen Schlüsseltabelleneintrags. In den weiteren Attributen werden die der Kennung zugeordneten Werte angegeben (in den meisten Fällen ist der Kennung nur ein „Langtext“ zugeordnet).

Für die meisten Schlüsseltabellen ist ein fester Wertekatalog vorgegeben, der Bestandteil der jeweiligen OKSTRA-Version ist und von dem nicht abgewichen darf. Eine dynamische Erweiterbarkeit besteht somit lediglich in einem technischen Sinne (man könnte Werte ergänzen, ohne die Struktur des Datenmodells zu ändern), nicht jedoch hinsichtlich der beabsichtigten Semantik.

2.2.1.2 Teilschemata und UML-Diagramme

Das OKSTRA-Kern Schema in der Version 2.017 teilt sich in mehrere Teilschemata, die im Folgenden kurz beschrieben werden:

Schema	Kommentar
Datentypen.xsd	Basisschema für elementaren und komplexe Datentypen
okstra.xsd	Hauptschema, das andere Schemata inkludiert
okstra-basis.xsd	Basisschema für die Typen „ObjectRefType“, „KeyValuePropertyType“, „OKSTRAMetaDaten“, „OKSTRAMetaDatenType“, „KoordinatenMetadaten“ und „KoordinatenMetadatenType“
okstra-typen.xsd	Basisschema für die Definition von Basiseinheiten (z. B. Kilometer)
S_Administration.xsd	Fachschema „Administration“
S_Allgemeine_Geometrieobjekte.xsd	Basisschema für Geometrie ohne Fachmodellierung, DGM und Punktwolken
S_Allgemeine_Mengenberechnung.xsd	Fachschema für Realisierung des Datenaustauschformates für die REB-VB 23.003 Ausgabe 2012
S_Allgemeine_Objekte.xsd	Basisschema für „allgemeine Objekte“
S_Arbeitsstelle_an_Strassen.xsd	Fachschema für Darstellung von Arbeitsstellen an Straßen
S_Bauliche_Strasseneigenschaften.xsd	Fachschema „Bauliche Straßeneigenschaften“
S_Bauwerke.xsd	Fachschema „Bauwerke“
S_Dynamische_Beschilderung.xsd	Fachschema „Dynamische Beschilderung“
S_Dynamische_Verkehrsdaten.xsd	Fachschema „Dynamische Verkehrsdaten“
S_Entwurf.xsd	Fachschema „Entwurf“
S_Flaechenmodell.xsd	Fachschema „Flächenmodell“
S_Grunderwerb.xsd	Fachschema für Grunderwerb
S_Hausnummern.xsd	Fachschema zur Darstellung von Hausnummern, Hausnummernbereichen und Hausnummernblöcken
S_Historisierung.xsd	Basisschema für Versionierung
S_Kataster.xsd	Fachschema Kataster
S_Kostenmanagement.xsd	Fachschema für Objektarten zur Darstellung von Daten zu Kostenermittlung und -management auf Basis der AKVS 2014
S_Kreuzungen.xsd	Fachschema für Kreuzungen klassifizierter Straßen mit anderen Verkehrswegen, Gewässern, Tierwechselln etc.
S_Landschaftsplanung.xsd	Fachschema für Landschaftsplanung
S_Lichtsignalanlage.xsd	Fachschema für Lichtsignalanlagen
S_Liegenschaftsverwaltung.xsd	Fachschema „Liegenschaftsverwaltung.“
S_Netzaenderungsprotokoll.xsd	Fachschema für das OKSTRA-Netzänderungsprotokoll, mit dem die Netzreferenzen von in sekundären Datenhaltungen befindlichen Fachdaten an Netzänderungen angepasst werden können
S_Oekologie.xsd	Fachschema für Objektarten aus der Ökologie
S_Organisation.xsd	Fachschema zur Darstellung von Organisationen und Zuständigkeiten
S_Projektressourcen.xsd	Fachschema „Projektressourcen“
S_REB_22013.xsd	Fachschema zur Realisierung des Datenformates der REB-VB 22.013.
S_Schwertransport.xsd	Fachschema zur Darstellung eines Schwertransports sowie der Parameter und der Ergebnisse der zur Genehmigung eines Schwertransports nötigen statischen Berechnungen an den zu überquerenden Bauwerken
S_Statische_Beschilderung.xsd	Fachschema statische Beschilderung
S_Strassenausstattungen.xsd	Fachschema „Straßenausstattungen“
S_Strassennetz.xsd	Fachschema „Straßennetz“
S_Strassenverzeichnis.xsd	Fachschema „Straßenverzeichnis“
S_Strassenzustandsdaten.xsd	Fachschema „Straßenzustandsdaten“
S_Topografie.xsd	Fachschema zur Darstellung topografischer Gegebenheiten.
S_Umfeldmessstelle.xsd	Fachschema Modell zur Darstellung von Umfeldmessstellen.

S_Unfall.xsd	Fachschemata „Unfall“
S_Verkehr.xsd	Fachschemata für den Themenbereich "Verkehr"
S_Vermessungspunkt.xsd	Fachschemata für die Darstellung von Vermessungspunkten
Schlussetabellen.xsd	Basisschemata für Schlüssel Tabellen des OKSTRA

Das OKSTRA-Schema enthält ganz allgemeine Basisschemata, die die Grundlage für Fachschemata bilden. So definiert beispielsweise das Basisschema „Datentypen“ grundlegende Datentypen, die anschließend von den verschiedenen Fachschemata (wie z. B. Fachschema „Entwurf“) genutzt werden können. Ähnlich verhält es sich auch z. B. mit dem Schema „Historisierung“. Dieses erlaubt es, Zustände eines Objektes zu unterschiedlichen Zeitpunkten zu erfassen und kann wiederum als Basis für die konkrete Fachmodellierung genutzt werden.

2.2.1.3 Ausschnittweise Betrachtung des Datenmodells

Im Folgenden werden Teile des OKSTRA-Datenmodells als UML-Diagramm dargestellt. Es wird darauf hingewiesen, dass es sich hier nur um kleine Ausschnitte handelt, da das OKSTRA-Modell sehr umfangreich ist. Die Ausschnitte wurden ausgewählt, um die Tiefe und den Detaillierungsgrad des OKSTRA-Modells zu illustrieren.

Im Basisschema „Allgemeine Geometrieobjekte“ finden sich allgemeine Objekte zur Beschreibung von Punkt-, Linien- und Flächenobjekten (siehe Abbildung 11). Diese werden für alle geometrischen Repräsentationen eingesetzt und machen ihrerseits Gebrauch von den zugehörigen GML-Datentypen. Nähere Ausführungen zur Definition und Nutzung geometrischer Informationen können Abschnitt 2.2.1.5 entnommen werden.

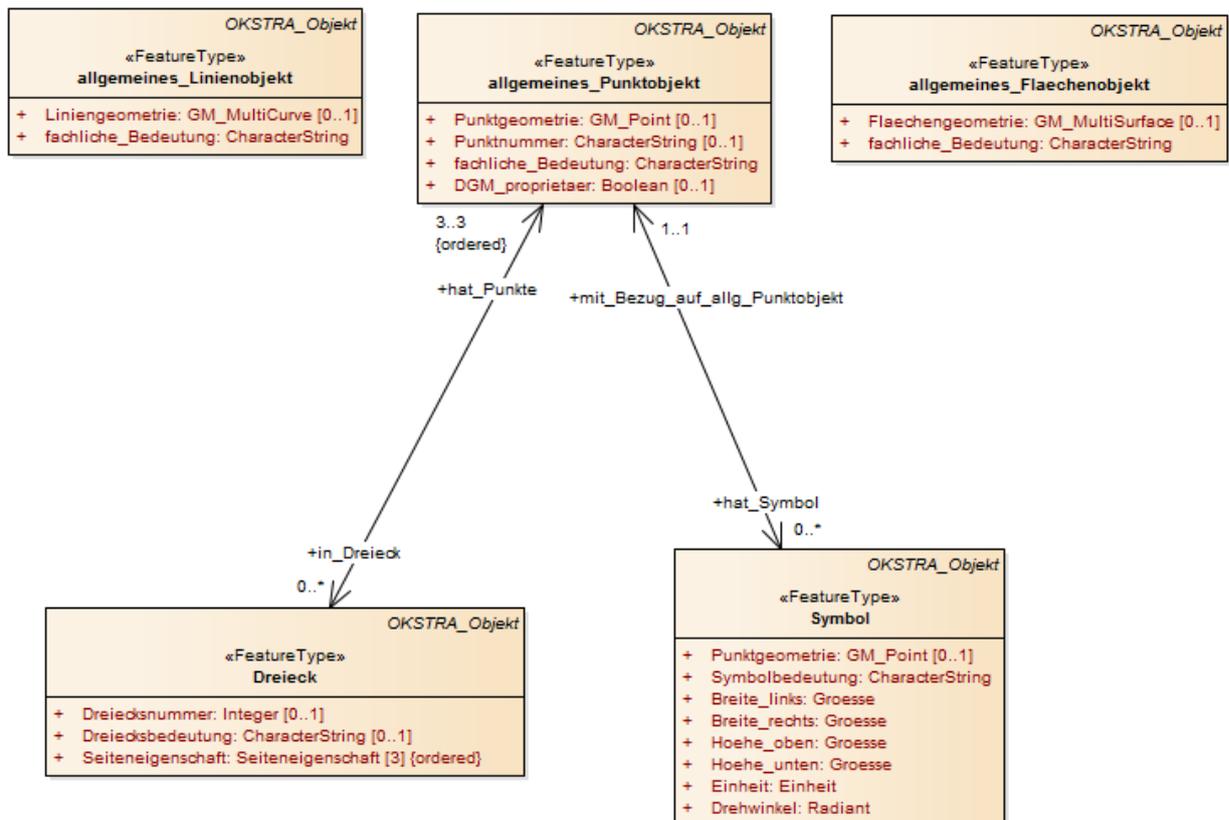


Abbildung 11: Allgemeine Geometrieobjekte des OKSTRA-Standards

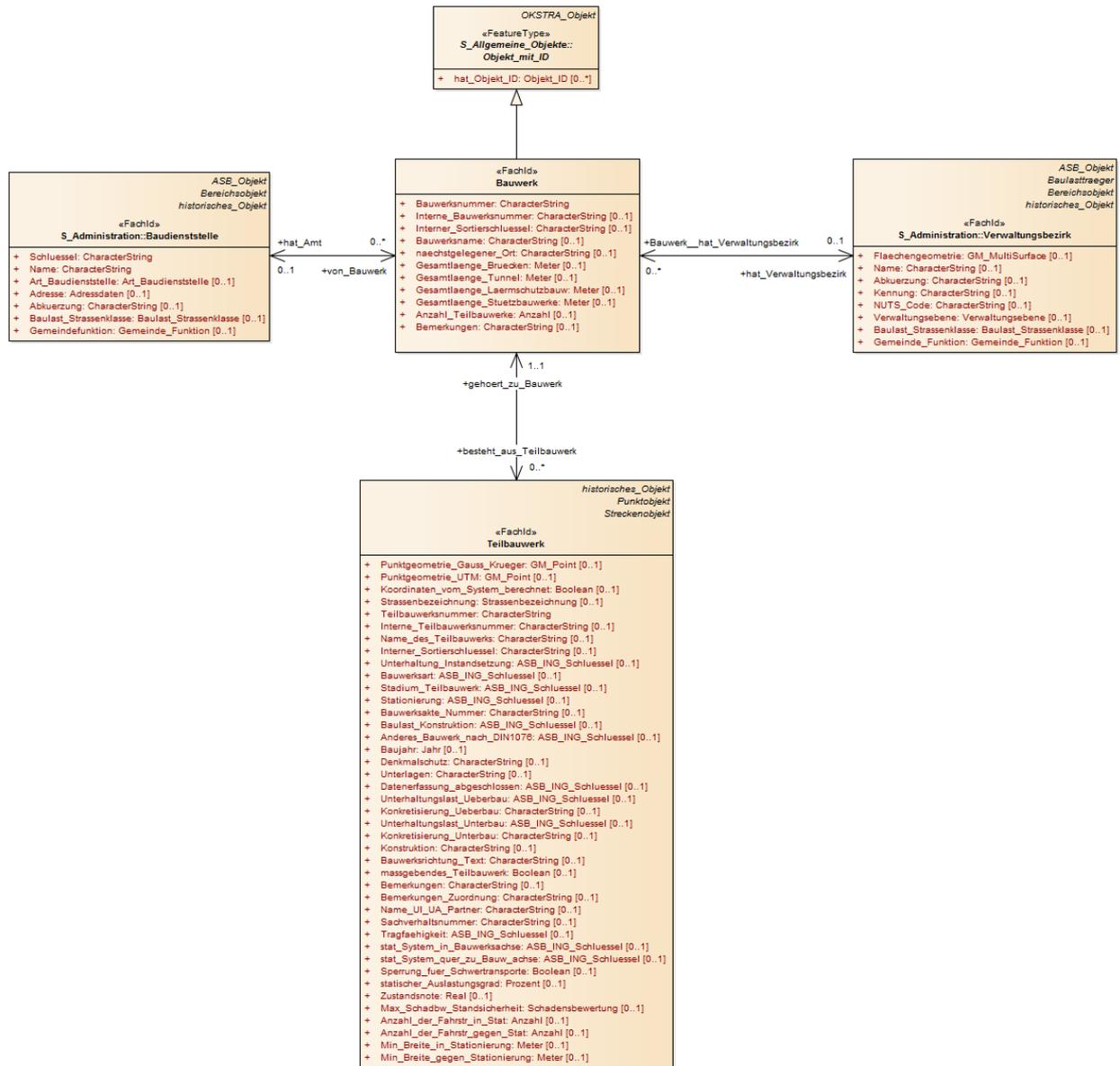


Abbildung 13: S_Bauwerk enthält die Informationen zur Beschreibung von Bauwerken wie Brücken und Tunneln

Das Teilschema *S_Bauwerk* enthält Datentypen zur Beschreibung von Ingenieurbauwerken wie Brücken und Tunneln, die Teile von Straßen sind (Abbildung 13). Das zentrale Element bildet dabei die Klasse *Bauwerk*, die allgemeine Verwaltungsinformationen bereitstellt und ihrerseits aus einer Menge von Teilbauwerken zusammengesetzt sein kann. Für jedes Teilbauwerk können eine Vielzahl von Informationen gespeichert werden, u.a. zur Konstruktion, zur Tragfähigkeit und zum Bauwerkszustand.

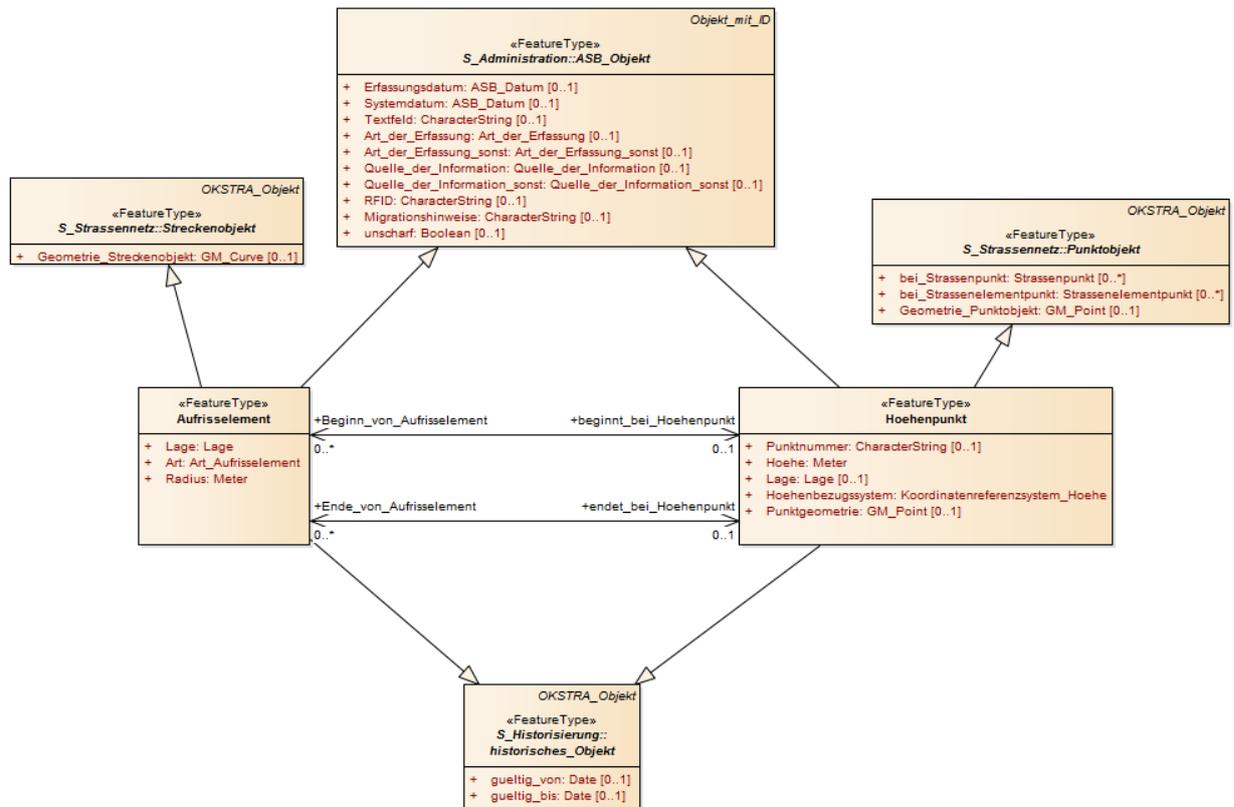


Abbildung 14: S-Administration: Die Klasse ASB-Objekt kapselt erhaltungsrelevante Informationen. Hier dargestellt ist die Verknüpfung mit Streckenobjekten.

Das Teilschema *S_Administration* beinhaltet wichtige verwaltungstechnische und erhaltungsrelevante Informationen. Die "Anweisung Straßeninformationsbank" (ASB) definiert eine Reihe von Eigenschaften, die bei allen in den Straßeninformationsbanken gehaltenen Objekten (SIB-Objekten) angegeben werden können. Im OKSTRA wurde zur Aufnahme dieser Eigenschaften der abstrakte Supertyp *ASB_Objekt* eingeführt. Alle auf der Grundlage der ASB modellierten OKSTRA-Objektarten erben von diesem Supertypen.

Abbildung 14 zeigt die Verknüpfung von *ASB_Objekt* mit Trassierungsobjekten, Abbildung 15 zeigt die Modellierung von Aufbauschichten des Straßenkörpers auf der Basis von *ASB_Objekt*.

Die UML-Diagramme illustrieren die Feingranularität und den hohen Informationsgehalt des OKSTRA-Datenmodells. Gleichzeitig ist anzumerken, dass eine Vielzahl von Attributen Textfelder sind, deren mögliche Belegung über Schlüssel Tabellen geregelt wird. Dies von der UML-Enumeration abweichende Modellierung erschwert die semantische Kopplung (Linking) mit anderen Datenmodellen.

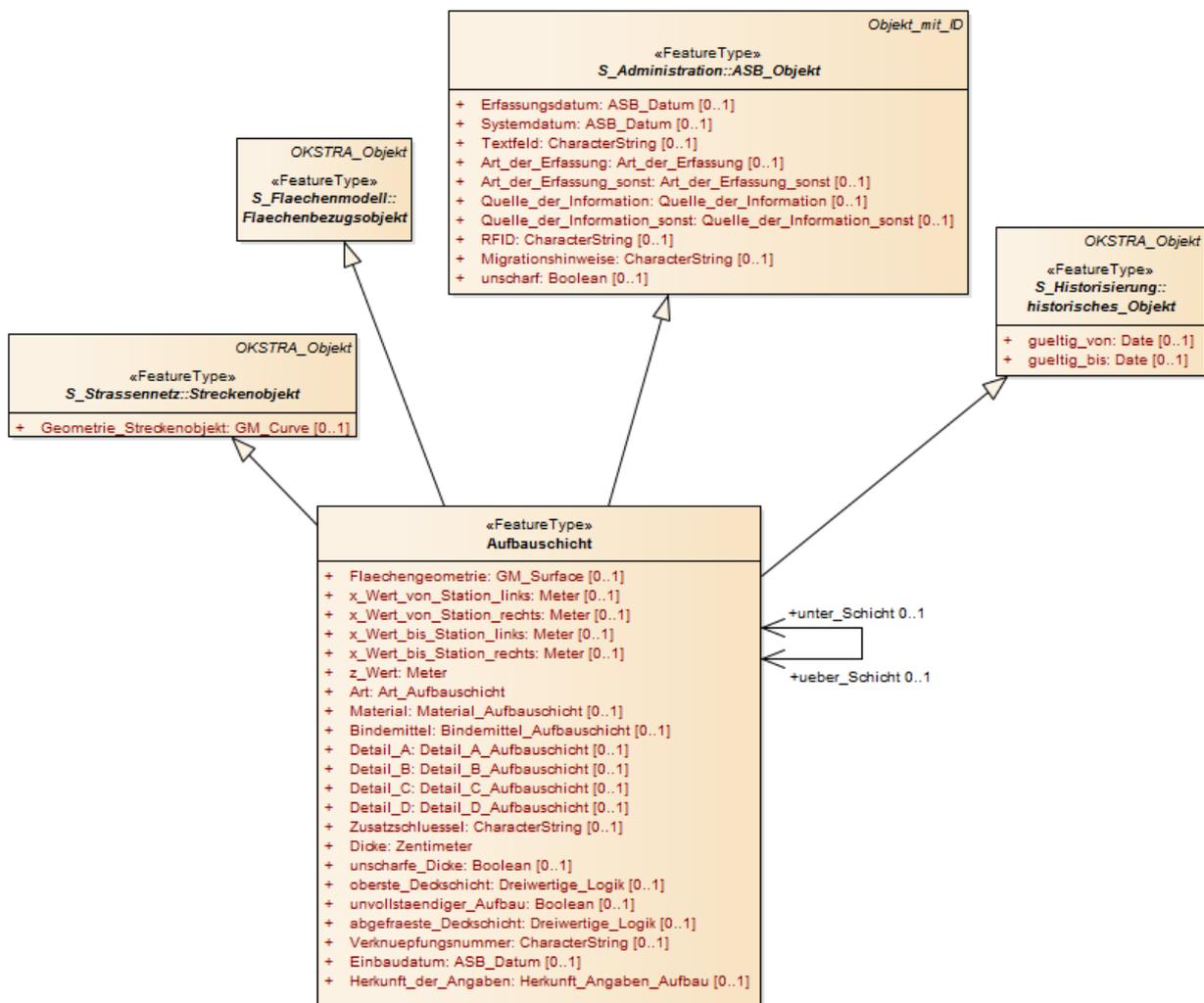


Abbildung 15: Die Darstellung der Aufbauschichten einer Straße

2.2.1.4 Analyse der semantischen Auszeichnungsqualität

Der OKSTRA-Standard ist mit seinen fast 14.000 Attributen wohl einer der umfangreichsten Standards seiner Art, die sich derzeit in Verwendung befinden. Im Vergleich mit CB-NL zeichnet sich OKSTRA durch eine sehr feine Granularität und einen hohen Detaillierungsgrad aus. Das Datenmodell ist sehr eng auf die Anforderungen von deutschen Verwaltungsbehörden und die deutsche Gesetzgebung zugeschnitten.

Um den OKSTRA-Standard flexibel mit möglichst geringem Aufwand anpassen zu können, werden Schlüsselstabellen genutzt. Dies erlaubt es, mit geringem Aufwand Änderungen in den Standard einzubringen (z.B. neue valide Attributwerte). Die meisten OKSTRA-Schlüsselstabellen haben feste Wertekataloge, die Bestandteil der jeweiligen OKSTRA-Version sind und gegen die geprüft werden kann. Es bleibt jedoch festzuhalten, dass die gewählte Vorgehensweise nicht der Standardmodellierung von Enumerationen in UML entspricht und damit auch nicht der Standardabbildung auf XML-Schema und sich damit Schwierigkeiten bei der Anwendung von Linked Data Methoden ergeben.

Gleiches gilt für die Fachbedeutungslisten, die im Zusammenhang mit allgemeinen Geometrieobjekten verwendet werden können. Dabei wird grob vereinfacht einem allgemeinen Geometrieobjekt eine eindeutige ID zugewiesen, deren Bedeutung in einer Fachbedeutungsliste näher spezifiziert ist. Die Komplexität dieser Auszeichnung erhöht sich dadurch, dass jedes Bundesland eigene Fachbedeutungslisten einbringen darf und diese auch in verschiedenen Versionen vorliegen dürfen. Damit ergibt sich die Gefahr, dass Daten zwischen den Bundesländern inkompatibel sind. Abbildung 16 zeigt eine Fachbedeutungsliste von Bayern in der Version 2.0.

	A	B	C	D	E	F	G	H	I
811	01.11.2006	#200.9130		KL-Mauer	Flaeche	20030			
812	01.11.2006	#200.9135		KL-Grenzpunkt grafisch	Punkt	20035			
813	01.11.2006	#200.9135		KL-Gem./Gemarkgr.n.ab.	Linie	20035			
814	01.11.2006	#200.9140		KL-Gebäudepunkt	Punkt	20040			
815	01.11.2006	#200.9140		KL-Gemarkungsgrenze	Linie	20040			
816	01.11.2006	#200.9140		KL-Treppe	Flaeche	20040			
817	01.11.2006	#200.9145		KL-Gemarkungsgr.n.ab.	Linie	20045			
818	01.11.2006	#200.9150		KL-Nutzungsartpunkt	Punkt	20050			
819	01.11.2006	#200.9150		KL-Gemeindegrenze	Linie	20050			
820	01.11.2006	#200.9155		KL-Gemeindegr.n.ab.	Linie	20055			
821	01.11.2006	#200.9160		KL-Sonstiger Punkt	Punkt	20060			
822	01.11.2006	#200.9160		KL-Flurstücksgrenze	Linie	20060			
823	01.11.2006	#200.9165		KL-Flurstücksgr.n.ab.	Linie	20065			
824	01.11.2006	#200.9170		KL-Höhenfestpunkt	Punkt	20070			
825	01.11.2006	#200.9170		KL-Gebäudeumriss	Linie	20070			
826	01.11.2006	#200.9175		KL-Gebäude-Detail	Linie	20075			
827	01.11.2006	#200.9180		KL-Gebäude-Durchfahrt	Linie	20080			
828	01.11.2006	#200.9190		KL-Nutzungsartgrenze	Linie	20090			
829	01.11.2006	#201.9100		KL-Gewässergrenze	Linie	20100			
830	01.11.2006	#201.9110		KL-Zaun	Linie	20110			
831	01.11.2006	#201.9120		KL-Mauer	Linie	20120			
832	01.11.2006	#201.9130		KL-Hecke	Linie	20130			
833	01.11.2006	#201.9140		KL-Fahrbahn,Gehweg,Gleis	Linie	20140			
834	01.11.2006	#201.9150		KL-Mauerlinie	Linie	20150			
835	01.11.2006	#201.9160		KL-Böschungskante,Grube	Linie	20160			
836	01.11.2006	#201.9170		KL-Durchlass	Linie	20170			
837	01.11.2006	#201.9180		KL-Brücke	Linie	20180			
838	01.11.2006	#201.9190		KL-Fußweg	Linie	20190			
839	01.11.2006	#205.9100		KL-Zugehörig.-Haken	Punkt	20500			

Abbildung 16: Fachbedeutungsliste für Bayern.

Das OKSTRA-Datenschema ist seit seiner initialen Entwicklung im Jahr 1999 bis zum heutigen Zeitpunkt stetig gewachsen. Beispielsweise hat die Anzahl von komplexen Typen (complex types) in den verschiedenen Versionen des OKSTRA-Standards zugenommen (siehe Abbildung 17).

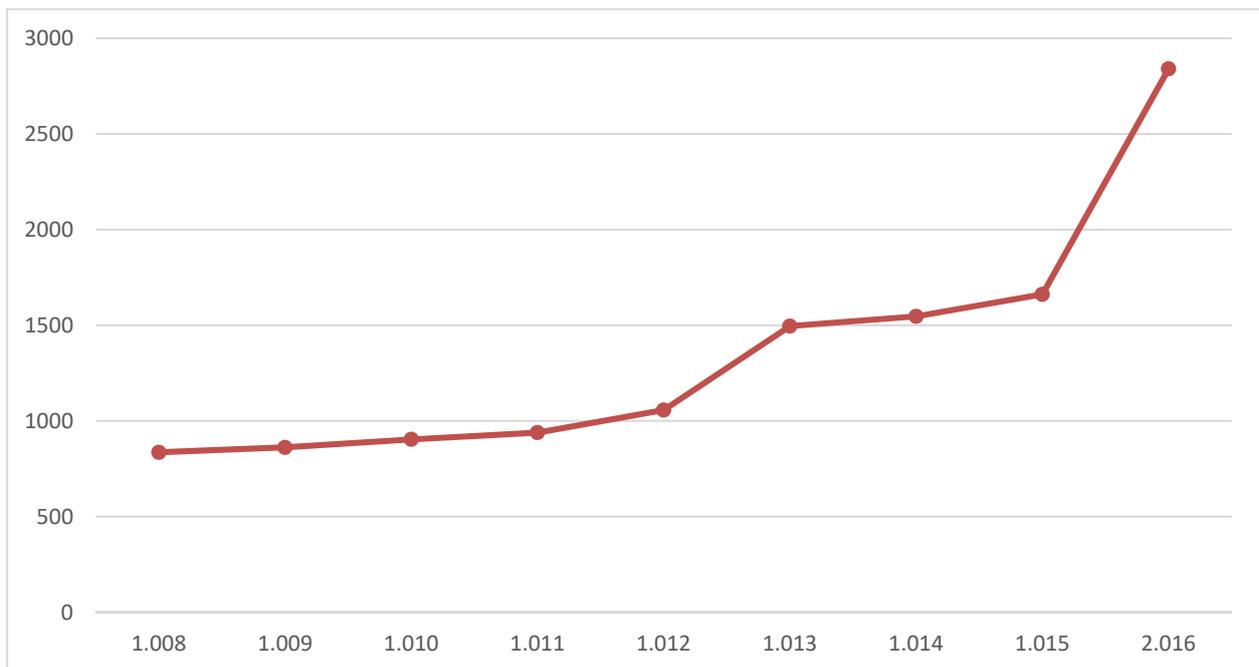


Abbildung 17: Zunehmende Anzahl von complex types in verschiedenen Versionen des OKSTRA-Standards

Die Anzahl der Attribute (siehe Abbildung 18), sowie die durchschnittliche Anzahl von Attributen pro Klasse (siehe Abbildung 19) hat ebenfalls in den verschiedenen Versionen des OKSTRA-Standards kontinuierlich über die Entwicklungszeit des Standards zugenommen.

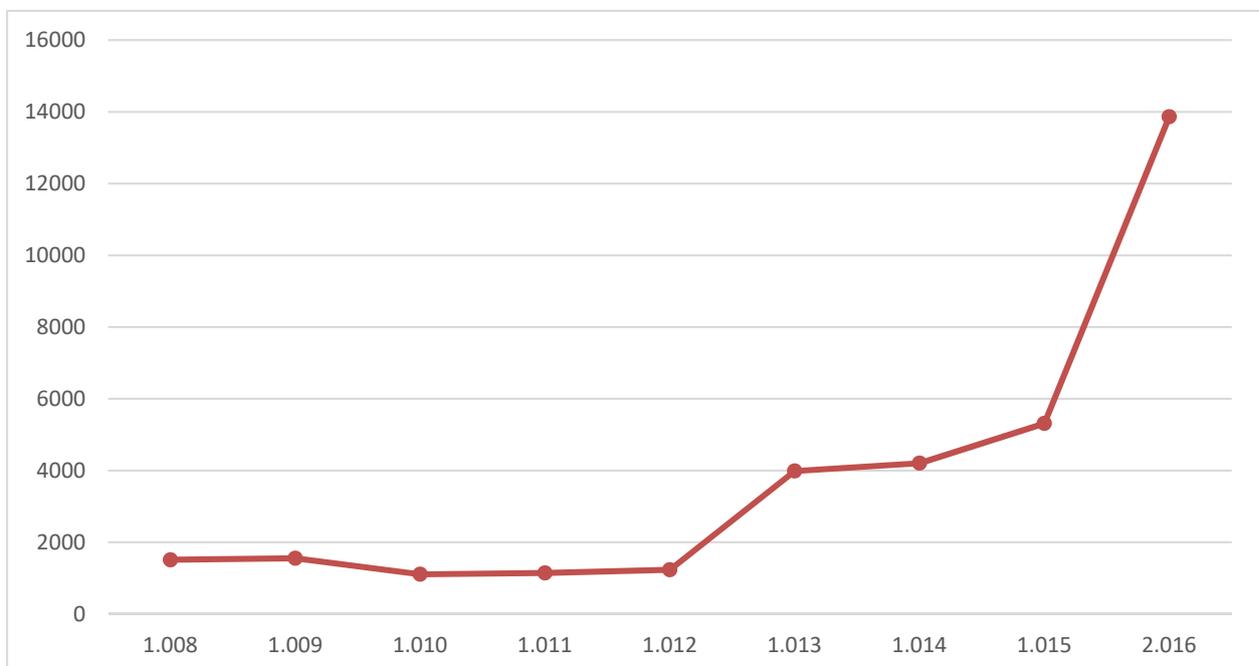


Abbildung 18: Anzahl von Attributen in verschiedenen Versionen des OKSTRA-Standards

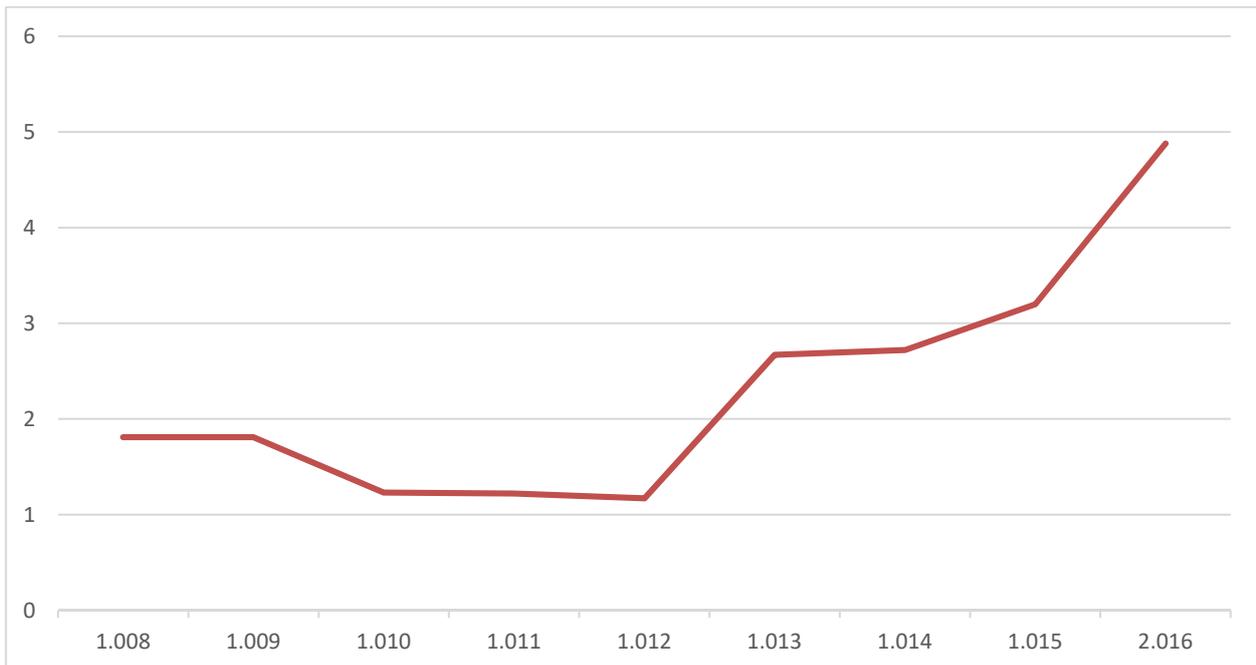


Abbildung 19: Durchschnittliche Anzahl von Attributen pro Klasse in verschiedenen Versionen des OKSTRA-Standards

Da nicht bei jedem Datenaustauschscenario alle Objekttypen des OKSTRA-Standards benötigt werden, sondern meistens nur eine sehr kleine Teilmenge, kann dieser auf die für einen konkreten Anwendungsfall zugeschnittene Menge von Objekten mithilfe sogenannter OKSTRA-Profile¹⁶ reduziert werden.

2.2.1.5 Analyse geometrischer Modellierungsprinzipien

Schema Allgemeine Geometrieobjekte

Im Basisschema „Allgemeine Geometrieobjekte“ finden sich allgemeine Objekte zur Beschreibung von Punkt-, Linien- und Flächenobjekten (siehe Abbildung 20).

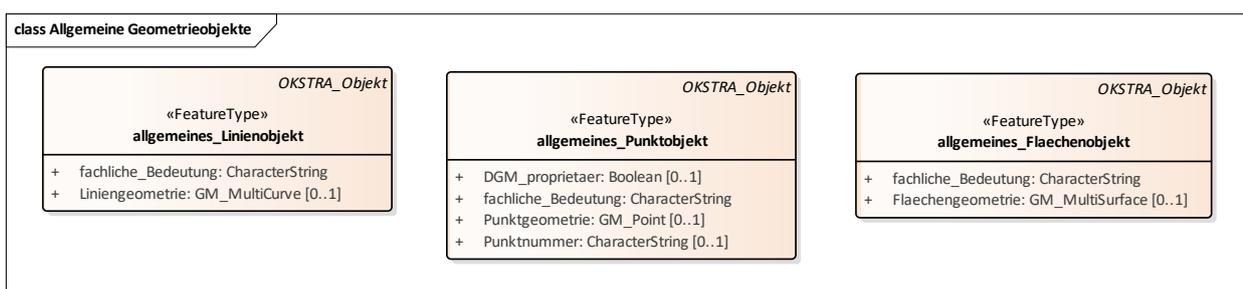


Abbildung 20: Allgemeine Geometrieobjekte

Diese stützen sich auf die GML-äquivalente Repräsentationen zur Geometrieabbildung (*GM_Point*, *GM_MultiCurve*, *GM_MultiSurface*). So hat das allgemeine Punktobjekt beispielsweise ein Attribut vom Typ *GM_Point*. Allgemeine Geometrieobjekte werden im Regelfall genutzt, um Geometrie für Objekte zu beschreiben, für die keine zugehörige Objektart in OKSTRA existiert.

Weiter befindet sich im Schema „Allgemeine Geometrieobjekte“ die Objektart DGM, die zur Repräsentation eines digitalen Geländemodells dient (Abbildung 21).

¹⁶ <http://www.okstra.de/docs/t-dokumente/t0009.pdf>

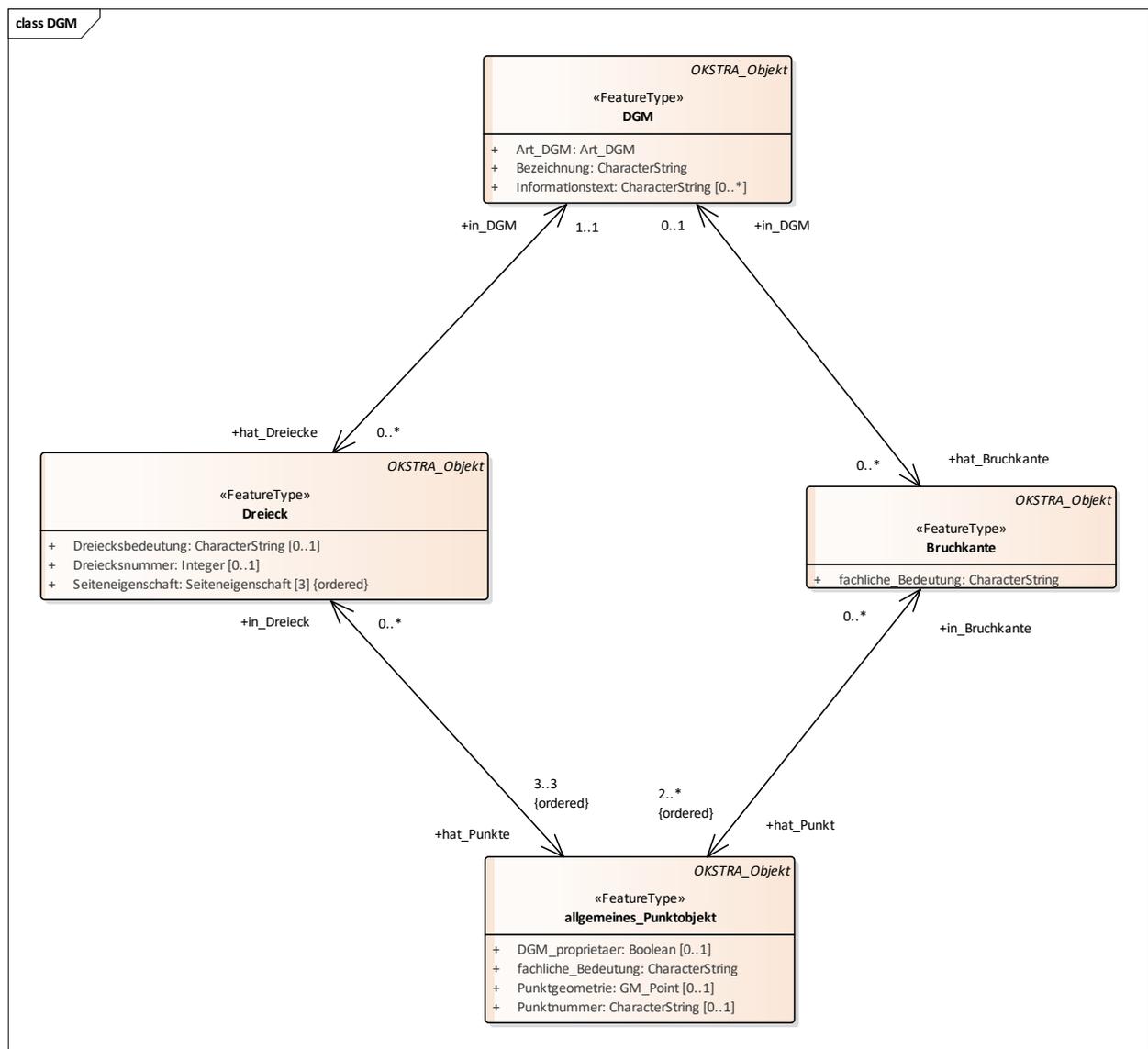


Abbildung 21: UML-Diagramm für die Modellierung des digitalen Geländemodells

An der Wurzel eines digitalen Geländemodells steht das Fachobjekt `DGM`. Dieses Fachobjekt besitzt ein Attribut `hat_Dreiecke`, das eine Sammlung von Fachobjekten vom Typ `Dreieck` speichert. Ein `Dreieck` selbst stützt sich wieder auf ein `allgemeines_Punktobjekt`, das wiederum die Geometrie als `GM_Point` abbildet. Abbildung 22 zeigt den Auszug einer Beispielinstantzdatei, die den Typ `allgemeines_Punktobjekt` verwendet.

```

<gml:featureMember>
  <okstra:allgemeines_Punktobjekt gml:id="allgemeines_Punktobjekt_359525">
    <okstra:Punktgeometrie>
      <gml:Point gml:id="Point.359526">
        <gml:pos srsDimension="3">1240.55 1297.057 187.211</gml:pos>
      </gml:Point>
    </okstra:Punktgeometrie>
    <okstra:fachliche_Bedeutung/>
    <okstra:in_Dreieck Objektklasse="Dreieck" xlink:href="#Dreieck_376877"/>
  </okstra:allgemeines_Punktobjekt>
</gml:featureMember>
  
```

Abbildung 22: Beispiel des Typs `allgemeines_Punktobjekt`

Schema Entwurf

Das Schema Entwurf dient der geometrischen Beschreibung einer Straße. Die Klasse `Achse` speichert dabei den Lageplan einer Trassierung. Sie besitzt ein Attribut mit dem Namen `hat_Achselement`, das eine sortierte Liste aller horizontalen Achselemente speichert (Lageplan). Achselemente können vom Typ Gerade, Kreisbogen und Klothoide sein. Jedes `Achselement` hat dabei einen Start- und Endpunkt, der durch einen `Achshauptpunkt` modelliert wird. Ein `Achshauptpunkt` wird dabei selbst geometrisch mittels des GML-Datentyps `GM_Point` beschrieben.

Frühere Versionen von OKSTRA nutzen noch nicht den GML-Standard zur Beschreibung von Geometrie beim Achsverlauf, sondern stellten ein eigenes Fachobjekt mit dem Namen `Punkt` bereit. Weitere Eigenschaften eines Lageplanelements, wie z. B. die Länge oder der Radius am Start- bzw. Endpunkt, werden durch eigens definierte Datentypen beschrieben. Beispielsweise wird für die genannten Attribute der Datentyp `Meter` verwendet, dessen Definition in Abbildung 23 zu sehen ist.

```
<complexType name="Meter">
  <simpleContent>
    <restriction base="gml:LengthType">
      <attribute name="uom" type="gml:UomSymbol" fixed="m" use="required"/>
    </restriction>
  </simpleContent>
</complexType>
```

Abbildung 23: Datentypdefinition von "Meter"

Die Gradiente wird mithilfe des Typs `Gradiente` durch eine in ihren Eckpunkten ausgerundete Polylinie beschrieben. Dabei gab es auch hier historisch gesehen kleinere Änderungen bei der Repräsentation eines Höhenplans. Die neuste Variante des OKSTRA-Standards (2.016, 2.017) nutzt das Fachobjekt `Grad_Koor`, das `Station` und `Hoehe` als Attribute beinhaltet und bei dem angegeben werden kann, ob eine Ausrundung existiert.

Abbildung 24 zeigt die Modellierung von Lage- und Höhenplan in einem UML-Klassendiagramm (nach der OKSTRA-Version 2.017).

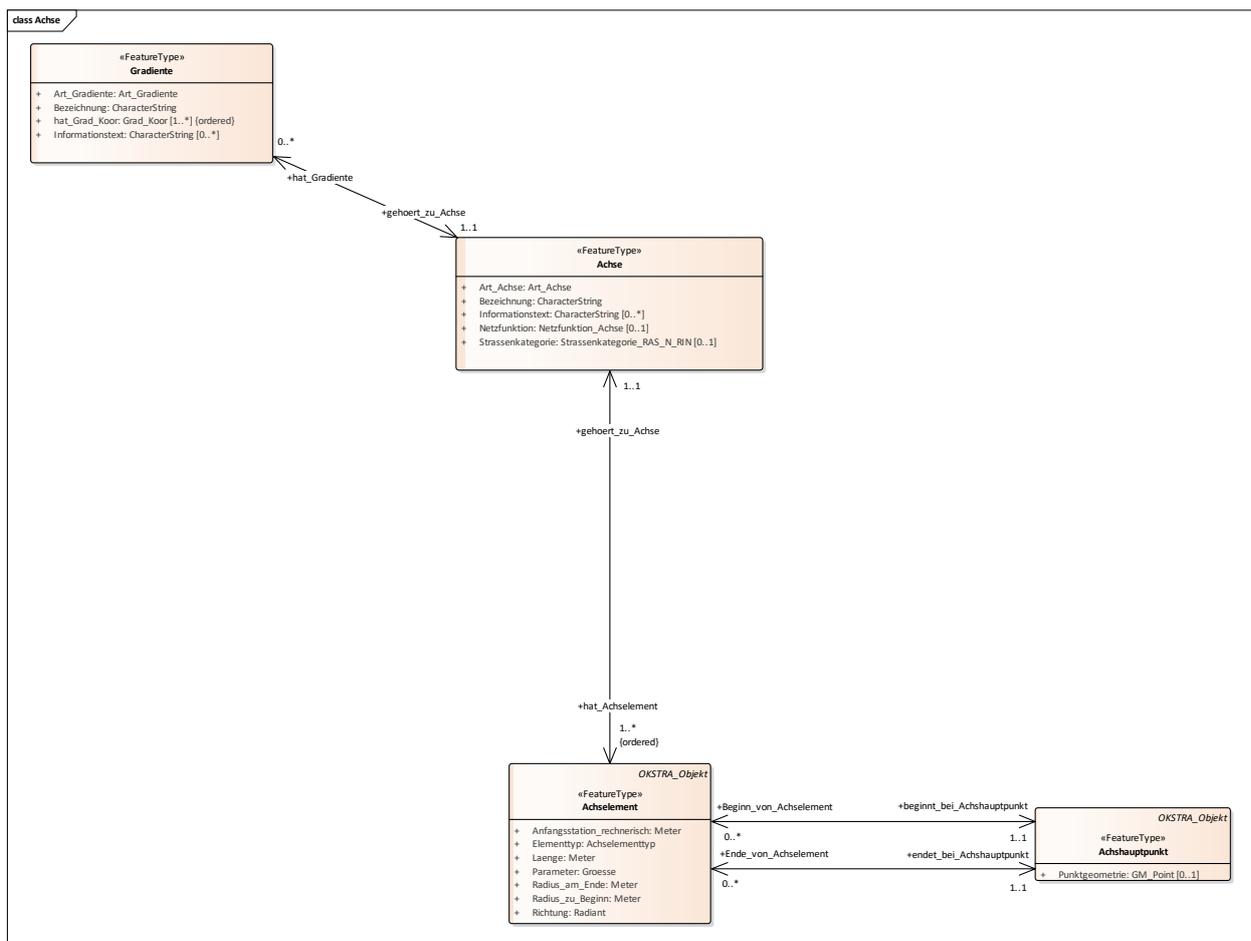


Abbildung 24: Modellierung von Lage- und Höhenplan in OKSTRA

Um Querprofildaten einer Trasse zu speichern, bietet OKSTRA das Fachobjekt `Querprofil` an. Dieses ist vergleichbar mit der `CrossSection` aus LandXML. Das Objekt Querprofil beschreibt alle Horizonte einer Straße in einem Querschnitt an einer bestimmten Station der zugeordneten Achse. Die einzelnen Horizonte des Querprofils werden über das Objekt `Profillinie` angegeben. Das Objekt `Profillinie` wiederum wird durch Objekte vom Typ `QP_Punkte` beschrieben. Diese Punkte sind über dem Abstand zur Achse und ihrer absoluten Höhe eindeutig bestimmt.

Zusätzlich können mit dem Objekt `Deckenbuch` die Spuren einer Straße geometrisch beschrieben werden (Abbildung 25). Mit jeder Spur muss ein Breitenband verknüpft und können optional ein Querneigungsband und ein Höhenzug angegeben werden. Für die geometrische Beschreibung des Breitenbands kommt der Datentyp `BR_Punkt` zum Einsatz, für das Querneigungsband `QN_Punkt` und für den Höhenzug `HZ_Punkt`, jeweils mit unterschiedlichen Attributen.

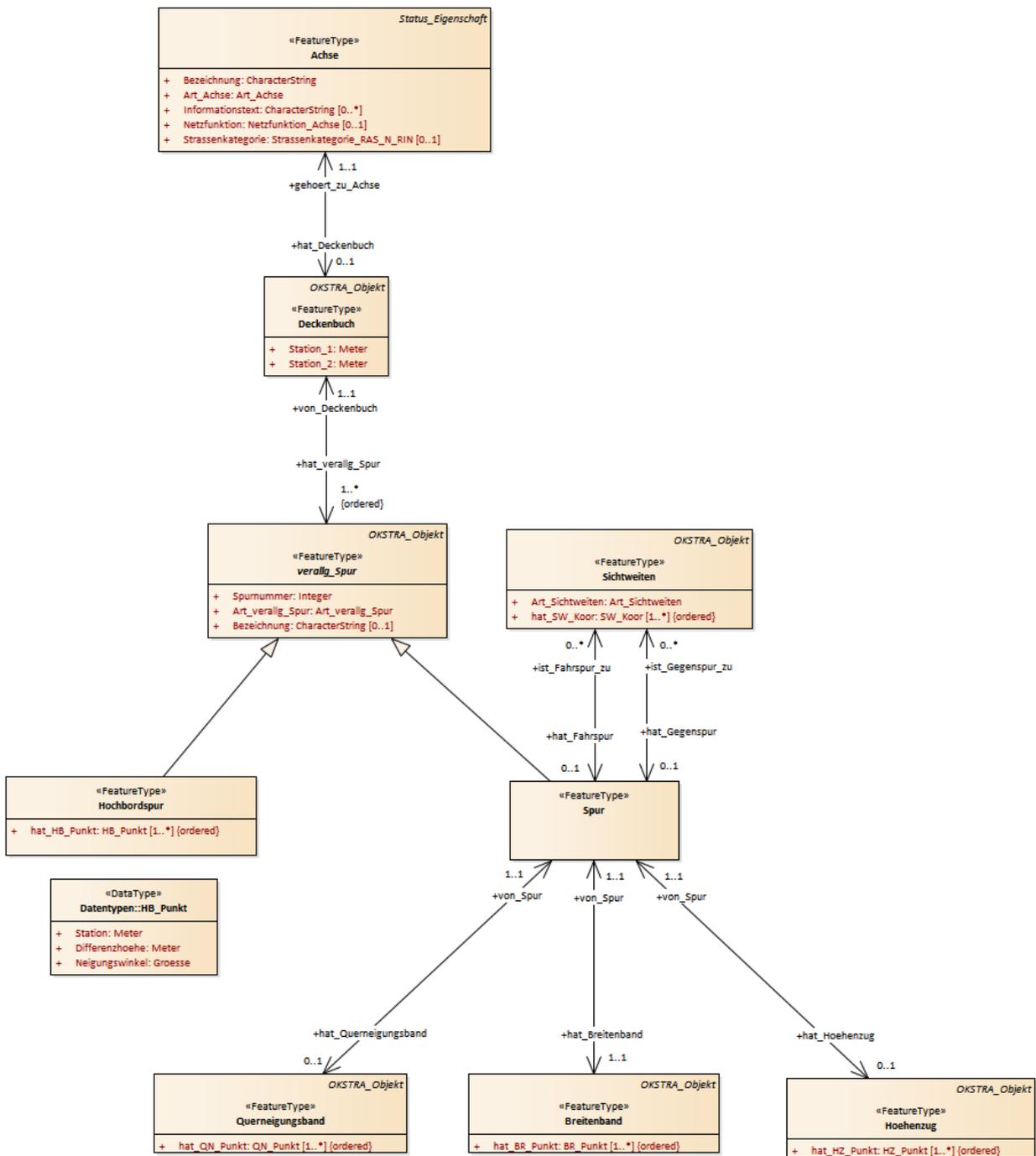


Abbildung 25: Modellierung des Deckenbuchs mit einer beliebigen Anzahl von Spuren, die jeweils Querneigungsband, Breitenband und Höhenzug aufweisen können

Weitere Fachschemata

Geometrieobjekte kommen in fast jedem Fachschema des OKSTRA-Standards vor. Beispielsweise findet sich im Schema „Unfall“ das Fachobjekt `Angaben_zum_Unfallort`, wo mittels eines `GM_Point`-Objektes der genaue Unfallort gespeichert werden kann. Im Schema „Topografie“ werden Böschungen, Isolinien, Mauern und andere Objekte mittels der Objektart `GM_Curve` beschrieben. Gebäudegrundrisse können als Flächengeometrie mittels `GM_Surface` beschrieben werden. Im Wesentlichen stützt sich der Großteil der geometrischen Beschreibung von OKSTRA auf die Typen nach ISO 19107 `GM_Point`, `GM_Curve` und `GM_Surface` (bzw. `GM_MultiPoint`, `GM_MultiCurve`, `GM_MultiSurface`). Darüber hinaus werden Höhenwerte, Stationierungen, Längen und beispielsweise Radien in der Entwurfsplanung mit eignen Datentypen, die wieder auf ISO19107-Basistypen basieren (wie z. B. „Meter“) beschrieben.

Zusammenfassung

Im Wesentlichen speichert OKSTRA 2.5D-Informationen, die durch Überlagerung zu einem 3D-Modell kombiniert werden können. Volumenkörper können nicht beschrieben werden. Das Entwurfsschema ist geprägt vom Paradigma der linearen Positionierung entlang der Achse (engl. linear referencing, ISO 19148). Ein Deckenbuch beschreibt die verschiedenen Fahrbahnsuren, die durch Offset-Werte definiert sind (*Breitenbezug*, *Breite*, usw.). Die Querneigung der Fahrbahnoberfläche wird ebenfalls durch 2D-Punkte definiert. Zwischen den verschiedenen Querschnitten kann durch Interpolation ein 3D-Modell hergeleitet werden. Allerdings hat die aktuelle Querschnittbeschreibung einige Mankos, z.B. ist sie weitgehend statisch und später durch Modellierungssoftware nur schwer modifizierbar. Möglichkeiten, diesen Umstand zu verbessern, wurden im Forschungsprojekt „Dynamische Querprofil“ untersucht¹⁷. Es muss allerdings festgehalten werden, dass sich das „Dynamische Querprofil“ bislang nicht durchgesetzt hat.

¹⁷ <http://www.okstra.de/docs/n-dokumente/n0046.pdf>

2.2.2 Analyse von CB-NL/OTL

2.2.2.1 CB-NL

Die "Conceptenbibliothek Nederland" (Konzeptbibliothek der Niederlande)¹⁸ ist als eine zentrale Sammlung von Konzepten und als Anknüpfungspunkt für bestehende und zukünftige Vokabulare, Klassifikationen, Ontologien und Datenmodelle angelegt. Die hierin modellierten Konzepte haben einen hohen Grad von Abstraktion und sind nicht als Datenmodell etwa zu einer umfassenden und detaillierten Modellierung von konkreten Bauvorhaben in Hoch- oder Tiefbau gedacht. Konkrete Eigenschaften und Attribute wie Dimensionierungen, geometrische Repräsentationen und Materialspezifikation sind hier nicht enthalten, und ein Vergleich mit OKSTRA ist damit nur in eingeschränktem Umfang möglich. Der zentrale Anwendungsfall der CB-NL ist die Verknüpfung unterschiedlicher Vokabulare - nationale und branchenspezifische strukturierte Vokabulare wie Klassifikationen oder Datenmodelle - in einem gemeinsamen Modell. Der Kern des CB-NL Modells ist dabei als Anker konzipiert, an dem Relationen zu anderen Vokabularen angehängen werden können, wie sie in Abbildung 26 übersichtsmäßig dargestellt ist.

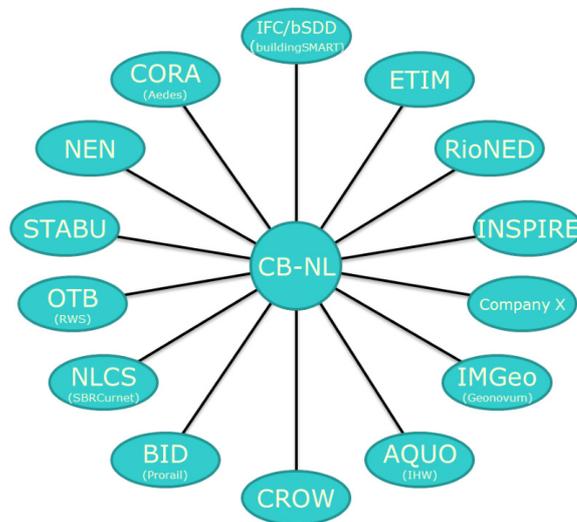


Abbildung 26: Konzeptionelle Darstellung des CB-NL Vokabulars als Ankerpunkt für verschiedene Fachvokabulare (Bakker & Mommers 2016)

In der CB-NL sind auf unterschiedlichen Betrachtungs- und Abstraktionsebenen grundlegende Konzepte erfasst, die in den folgenden Hauptkategorien unter dem Konzept "Objekt" in einem hierarchischen Klassenbaum mit teils mehreren Elternknoten (Mehrfachvererbung) eingeordnet sind:

- Abstrakte Objekte (z.B. Technik -> Bauingenieurwesen -> Straßenbau)
- Aktivitäten (z.B. Aktion->Überbrücken)
- Physische Objekte (z.B. Gebautes Objekt->Konstruktion->Straße)
- Ereignis (z.B. Brand)
- Informationsobjekte (z.B. Spezifikation->Anforderung)
- Organisationen (z.B. Polizei)
- Prozesse (z.B. Renovierung)
- Räumliche Objekte (z.B. Funktionsraum->Korridor->Weg->Spur->Busspur)
- Ströme (z.B. Materialstrom->Verkehr)
- Systeme (z.B. Netzwerke->Straßennetz)

¹⁸ <http://public.cbnl.org/61>

Das Gesamtvokabular ist in drei Hauptkategorien in Form von Subgraphen (ähnlich XML Namespaces) gegliedert:

1. Kern (CBLN Core) zur Erfassung konkreter Konzeptdefinitionen
2. Oberkonzepte (CBNL Top) zu Erfassung allgemeiner Abstraktionen
3. Metainformationen (CBNL Meta) zur Erfassung von Metadaten wie Autoren und Erstellungsdaten

Vielen der modellierten Konzepte sind Relationen zugeordnet, die nur eingeschränkt mit Objektbeziehungen und Attributen in Datenmodellen wie dem OKSTRA vergleichbar sind. Ein Beispiel hierfür ist das Konzept "Strasse", dem die folgenden Eigenschaften zugeordnet sind, die als Unterklassen-Restriktionen in OWL modelliert sind (näheres hierzu siehe Abschnitt 2.3 dieses Dokumentes): Jede Instanz des Konzeptes "Strasse" hat die Eigenschaft "hatAnwendung" die auf den Wertebereich der Klassen "Strassenverkehr" beschränkt ist. Konkrete Datentypen, etwa um Ergebnisse von Messungen aufzunehmen, sind nicht vorhanden.

Um eine Vergleichbarkeit bezüglich des Detaillierungsgrades z.B. mit OKSTRA zu ermöglichen, ist in Tabelle 1 die Anzahl der Konzepte der einzelnen Hauptkategorien sowie ihrer Relations-Restriktionen erfasst.

Graph	Anzahl Klassen	Davon mit Englischen Bezeichnungen	Relations-Restriktionen
CBNL Core	3452	1683	2589
CBNL Top	86	86	0
CBNL Meta	5	3	0

Tabelle 1: Übersicht über die Anzahl einzelner Klassen und Konzepte

In diesen Mengenangaben sind jeweils Verknüpfungen mit Konzepten aus anderen Vokabularen enthalten. Wie in Tabelle 2 aufgelistet, sind derzeit eine Reihe unterschiedlicher Konzepte aus anderen Vokabularen wie etwa Klassifikationen verknüpft. Im Einzelnen handelt es sich dabei um:

- die Object Type Library von Rijkswaterstaat (OTL RWS)¹⁹, siehe auch weiter unten
- NEN 2767-4, der Niederländischen Norm zur Zustandserfassung von Infrastruktur²⁰, die eine detaillierte Klassifikationsstruktur infrastruktureller Objekte ("nass" und "trocken", d.h. Wasser und Landinfrastruktur) enthält und ähnlich der deutschen DIN 276 für den Hochbau als Ordnungsstruktur vielfältige Anwendung findet
- Informatiemodel Geografie (IMGeo)²¹, einem Erweiterungskatalog für das CityGML Model²² mit Hilfe von Application Domain Extensions (ADE), die die Grundlage für niederländischen Taxonomie Basisregistratie Grootchalige Topografie (BGT)²³ bildet, siehe auch (Stoter, Beetz et al 2013)
- NL/SfB²⁴, die niederländische Variante der ursprünglich in Schweden entwickelten Klassifikationsstruktur für Bauten (Giertz 1982)

¹⁹ <https://otl.rws.nl/publicatieomgeving/#/>

²⁰ NEN 2767-4, Conditiemeting Infrastructuur; Engl: NEN 2767-4-1:2011 Condition assessment - Part 4: Infrastructure - Part 1: Methods, erhältlich auf <https://www.nen.nl/NEN-Shop-2/NEN-2767412011-nl.htm>, siehe auch <https://www.nen2767-4.nl/>

²¹ <http://www.geonovum.nl/wegwijzer/standaarden/gegevenscatalogus-imgeo-versie-211>

²² <https://www.citygml.org/>

²³ <https://bgtweb.pleio.nl/overdebg>

²⁴ <http://www.stabu.org/diensten/nlsfb/> und https://www.stabu.nl/mijnstabu/files/NL-SfB_BNA_Boek_2005-ISBN-10%2090-807626-3-6.pdf

- ElektroTechnisches InformationsModell (ETIM), eine international standardisierte Klassifikationsstruktur zur Beschreibung von technischen Produkten vor allem aus der Elektrotechnik²⁵

Vokabular	Anzahl Klassen
OTL RWS	57
NEN 2767-4	590
IMGeo	141
ETIM	491
NL/SfB	293
Gesamt	1572

Tabelle 2: Anzahl von Konzepten aus anderen Vokabularen, die in der CB-NL verknüpft sind.

Um eine Übersicht über den Grad der Spezialisierung der einzelnen Konzeptsubgraphen zu bekommen, sind in Tabelle 3 die Anzahlen der jeweiligen Subklassen von Konzepten erfasst. So finden sich unterhalb der weiter oben aufgelisteten 10 Hauptkonzepte der Ebene 1 ("Abstraktes Objekt", "Physisches Objekt" etc.), 77 Subklassen (in den Beispielen bspw. "Technik" und "Gebautes Objekt")

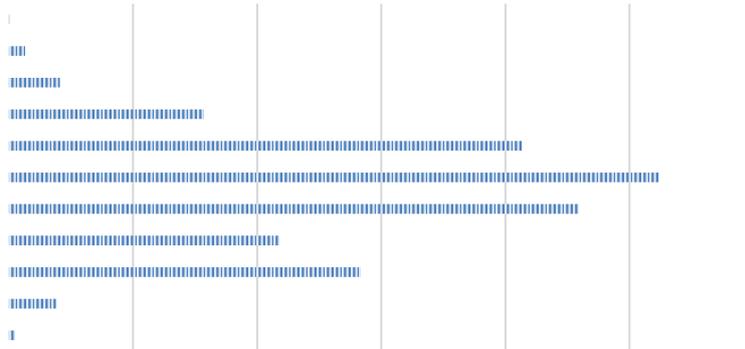


Tabelle 3: Detailtiefen des CB-NL Konzeptgraphen entlang der Subklassenbeziehungsachsen. Je höher die Ebene, desto spezialisierter das Konzept

2.2.2.2 RWS OTL

Die Object Type Library (OTL) von Rijkswaterstaat²⁶ ist im Gegensatz zur allgemeinen CB-NL darauf ausgelegt, konkrete Eigenschaften zu erfassen, mit denen beispielsweise funktionale Ausschreibungen, Vorgaben und grundlegende Systembäume einer Reihe unterschiedlicher Infrastrukturarten beschrieben werden können.

²⁵ <http://www.etim-international.com/classification/introduction>

²⁶ <https://otl.rws.nl/publicatieomgeving/#/>

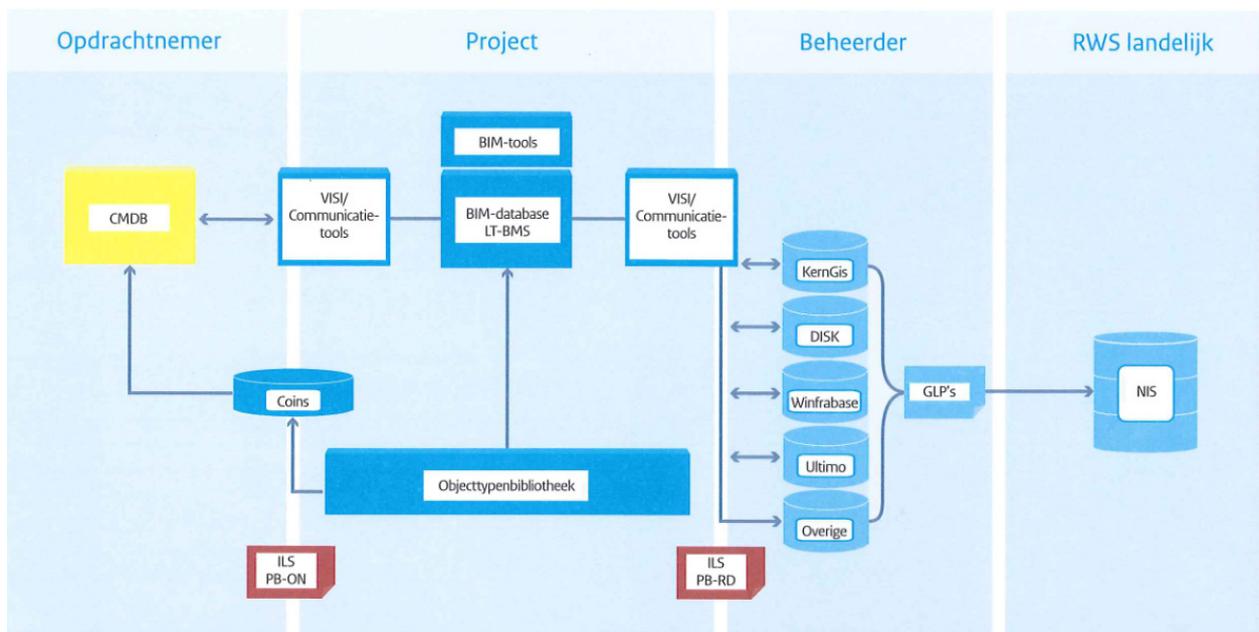


Abbildung 27: Übersicht über die Basisbestandteile der ILS (Quelle: RWS 2013)

Die RWS OTL ist Teil des Informationslieferungshandbuchs (IHL)²⁷²⁸ (niederländisch: Informatieleveringspecificatie ILS), dargestellt in Abbildung 27, die in ihrer ersten konkreten Ausprägung in Form der "BIM Basis ILS/IHL" eine Reihe von Standardinformations-Austauschformaten und Protokollen festlegt, um den reibungslosen Ablauf in allen Phasen des Bauens zu gewährleisten. Neben der OTL sind hier auch der COINS²⁹ Standard als Containerformat für strukturierte Verknüpfung verschiedener Datenbestände und deren Inhalte, das buildingSMART Information Delivery Manual (IDM)³⁰ sowie das Datenmodell der Industry Foundation Classes (IFC) aufgenommen. Die ILS Initiative stellt ebenfalls Anleitungen für konkrete BIM-Softwarepakete (Revit, Allplan, Tekla etc.) zur Verfügung, um die vorgestellte Verfahrensweise praktisch anwenden zu können. Das Basisprinzip beruht auf der geometrischen Produktmodellierung in Form von IFC, der Auszeichnung der Objekte mit einer OTL, der Zusammenstellung der Datensätze mittels eines COINS-Containers und dem Versenden von Änderungswünschen (Change Requests) mit Hilfe des niederländischen VISI-Protokolls.

Die RWS OTL trennt bei der Beschreibung von Straßenwegen strikt zwischen funktionalen und physischen Objekten. Abbildung 28 zeigt die Dekompositionsstruktur auf funktionaler und auf physischer Seite und die Verknüpfungen. Wie dort dargestellt, unterscheidet die RWS OTL 7 Detaillierungsniveaus. Die Niveaus 4-7 der physischen Beschreibung haben dabei die folgende Bedeutung (Abbildung 29):

- Niveau 4: Managementobjekt
- Niveau 5: Element
- Niveau 6: Bauteil
- Niveau 7: Unterbauteil

²⁷ <http://www.bimloket.nl/BIMbasisILS>

²⁸ http://www.bimloket.nl/upload/documents/downloads/BIMbasisILS/BIM%20basis%20ILS_infographicA4_German.pdf

²⁹ <http://www.coinsweb.nl/wiki2/>

³⁰ <http://iug.buildingsmart.org/idms/>

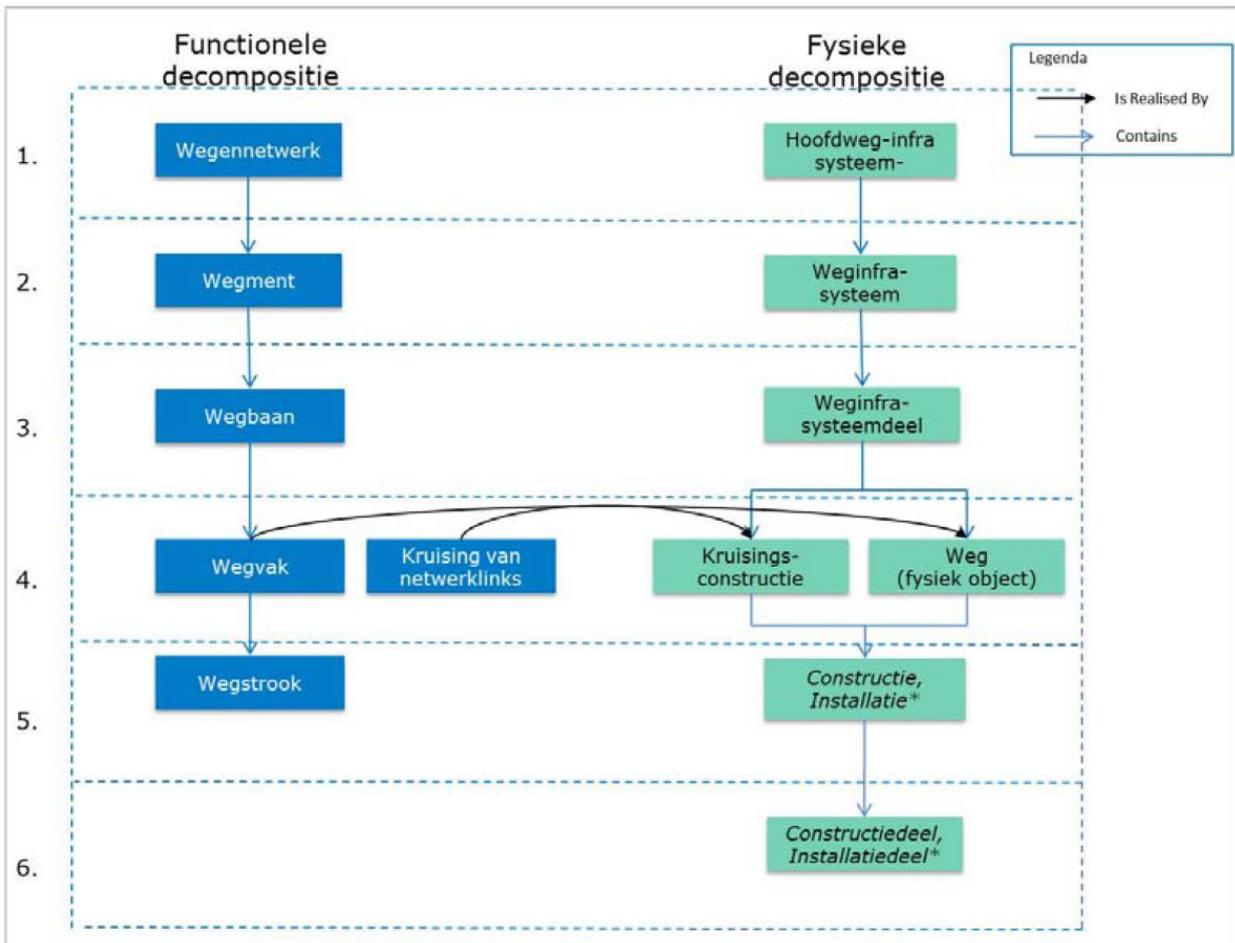


Abbildung 28: Funktionale Dekomposition und physische Dekomposition von Straßenwegen in der OTL (Quelle: RWS 2017)

Niveau 4 Beheerobject	Niveau 5 Element	Niveau 6 Bouwdeel
Weg (fysiek object)	Bovenbouw (weg)	Deklaag (weg), slijtlaag (weg), tussenlaag (weg), onderlaag (weg), fundering (weg)
	Onderbouw (weg)	

Abbildung 29: Die unterschiedliche Auflösung der Beschreibung des Straßenoberbaus auf den Detaillierungsebenen 4, 5 und 6 (Quelle: RWS 2017)

Die Niveaus 4-7 orientieren sich an der niederländischen Norm NEN2767-4 und übernehmen die dort festgelegten Begrifflichkeiten.

Zur Verdeutlichung des Detaillierungsgrads sind in Abbildung 30 die Komponenten des Straßenoberbaus nach RWS OTL dargestellt. Abbildung 31 stellt die Elemente der Straßenbeschilderung dar.

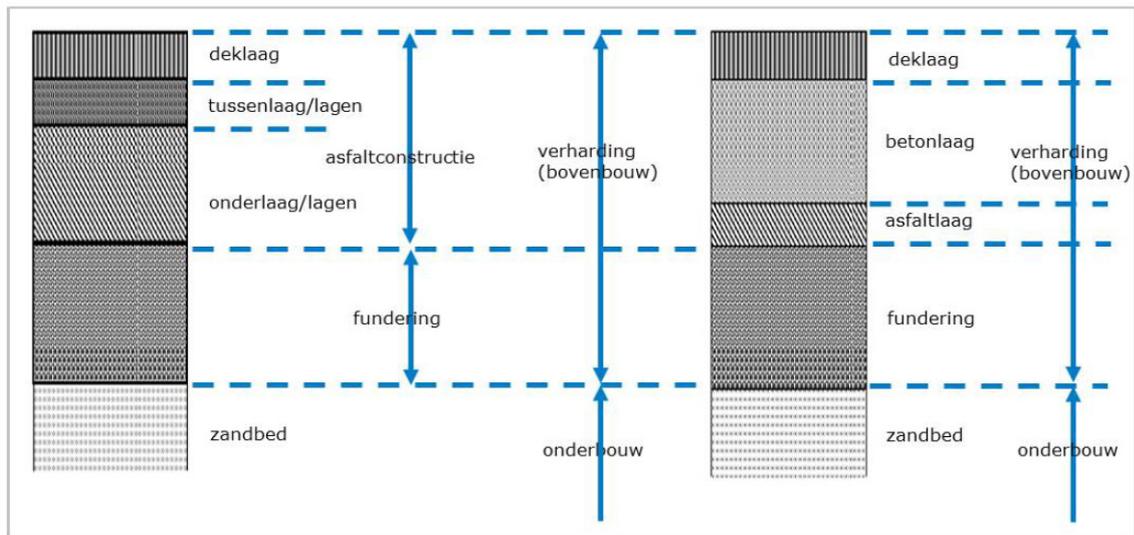


Abbildung 30: Komponenten des Straßenoberbaus nach RWS OTL: links Asphaltkonstruktion, rechts Betonkonstruktion (Quelle: RWS 2017)

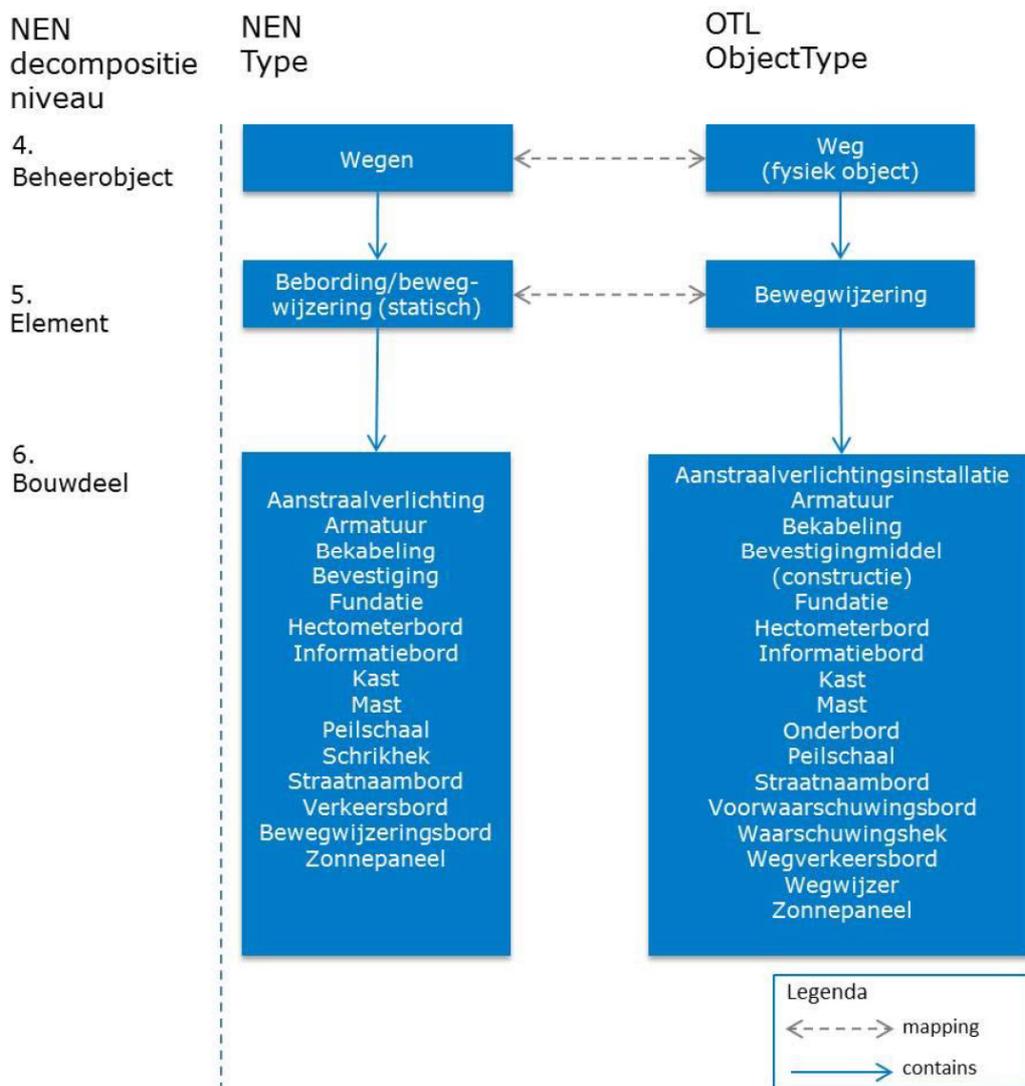


Abbildung 31: Elemente der Straßenbeschilderung: Links die nach NEN definierten Elemente, rechts die Konzepte der OTL (Quelle: RWS 2017)

Abbildung 32 und Abbildung 33 geben die Online-Dokumentation des Konzepts Bovenbouw (deutsch: Straßenoberbau) wieder.

BOVENBOUW (WEG)

OB00016

Informatie
Eigenschappen
Relaties
Mappings
Boomdiagram
Graafdiagram

otl-1.6
3 jaar geleden gewijzigd
PhysicalObjectType

— **Model:** OTL Conceptueel model
 — **Views:** otl-nen-bouwdeel, otl-nen-element

Definitie

Constructie bestaande uit een of meer verhardingslagen, om verkeer over het terrein mogelijk te maken.

Meta informatie

AliasName Verharding

Description De bovenbouw is in het horizontale vlak opgebouwd uit een of meerdere verharde banen (verkeersdragende banen), in het verticale vlak uit een of meer verhardingslagen.
 In de NEN 2767-4 worden zeven soorten Verharding genoemd, maar bestaat er geen gemeenschappelijk supertype Verharding.

Begrenzing

De bovenzijde van de component Bovenbouw wordt begrensd door bovenzijde deklaag.
 De onderzijde van de component Bovenbouw wordt begrensd door bovenzijde zandbed (= bovenzijde Onderbouw) of bovenzijde kunstwerk.
 Aan de zijkanten wordt de component Bovenbouw begrensd door bermen of schampkanten van een kunstwerk.

— Grens Bovenbouw - kunstwerk

Name in EN Paving

Abbildung 32: Beispiel aus der Online-Dokumentation von RWS OTL: Das Konzept *Bovenbouw* (deutsch: *Straßenoberbau*), hier die textuelle und graphische Beschreibung

BOVENBOUW (WEG)
OB00016

Informatie
Eigenschappen
Relaties
Mappings
Boomdiagram
Graafdiagram
↗

- aantal lagen (Bovenbouw (weg) bestaat uit aantal lagen)
- BMS identifier (Bovenbouw (weg) heeft kenmerk BMS identifier)
- Externe identifier (Bovenbouw (weg) heeft kenmerk Externe identifier)
- Locatie (Bovenbouw (weg) heeft kenmerk Locatie)
- Omschrijving (Bovenbouw (weg) heeft kenmerk Omschrijving)
- Type onderbouw (Bovenbouw (weg) heeft kenmerk Type onderbouw)
- verkeersklasse (Bovenbouw (weg) heeft kenmerk verkeersklasse)
- Wegtype (beheer) (Bovenbouw (weg) heeft kenmerk Wegtype (beheer)) ▼
- Zijde (Bovenbouw (weg) heeft kenmerk Zijde) ▼

Abbildung 33: Beispiel aus der Online-Dokumentation von RWS OTL: Die Eigenschaften des Konzepts *Bovenbouw*

Die RWS OTL besteht aus insgesamt 7.482 Konzept-Klassen als Unterklassen des `owl:Thing` Konzeptes von denen 2.010 auch mit einer englischsprachigen Auszeichnung (`rdfs:Label`) versehen sind. 10.681 Objektrelationen (`owl:ObjectProperty`) setzen diese neben den 22.242 Spezialisierungsbeziehungen (`rdfs:SubclassOf`³¹) zueinander in Beziehung. Diese Objektrelationen stammen aus der COINS-Ontologie und erfassen unter anderem mereonomische Teil-Ganzes Beziehungen (`coins-base:partOf`, `coins-base:ContainsRelation`), Verbindungen (`coins-base:hasConnection`) und Eigenschaften (`coins-base:hasProperties`). Beispielhaft sind in Abbildung 34 und Abbildung 35 die Beziehungen des Konzeptes "Spur" (einer Fahrbahn, niederl. "Wegstrook") wiedergeben.

³¹ hier sind nur die direkten `subClassOf` Beziehungen zwischen Klassen gezählt, keine Äquivalenzklassen etc.

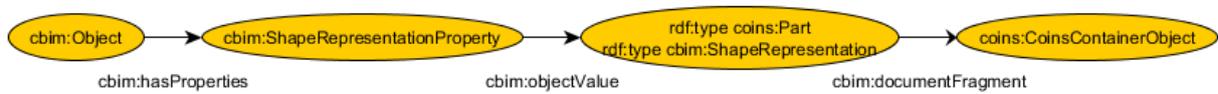


Abbildung 36: Verknüpfung einer durch die RWS-OTL ausgezeichneten Objektinstanz mit einer geometrischen IFC-Repräsentation

2.2.3 Vergleich der Anwendungsgebiete von OKSTRA und CB-NL

Der OKSTRA dient der Unterstützung der Prozesse für die Verwaltung und für den Bau von Bundesfernstraßen. Die Anwendungsgebiete des OKSTRA umfassen u.a.:

- den Straßenentwurf,
- den Grunderwerb,
- die Zustandsbewertung,
- das Kostenmanagement,
- die Landschaftsplanung,
- das Erheben von Verkehrs- und Unfalldaten und
- die Planung von Schwertransporten.

CB-NL dient hingegen als grundlegende Ontologie zur Verknüpfung unterschiedlicher Klassifizierungs- und Auszeichnungssysteme in den Niederlanden. Folglich sind für CB-NL die Anwendungsgebiete deutlich weniger scharf definiert als für OKSTRA. RWS OTL hat als wesentliches Anwendungsgebiet die Bestandsverwaltung (engl. Asset management), der Straßenentwurf ist aber weitgehend ausgenommen.

Nach Ansicht der Verfasser gibt es die größten Überlappungen in den folgenden Bereichen:

- Straßennetzwerk
- Kreuzungen
- Straßenaufbau (Bauliche Straßeneigenschaften)
- Beschilderung

In allen anderen Bereichen gibt es nur kleine bzw. keine Überlappungen.

Beispielhaft soll im Folgenden die Modellierungsmöglichkeiten in Bezug auf die Trassierung untersucht und verglichen werden. Das Entwurfsschema des OKSTRA-Standards kennt als Elemente des Lageplans gerade Liniensegmente, Klothoiden und Kreisbögen. Für eine Klothoide wird unter anderem der Startradius, Endradius der Klothoidenparameter, die Startrichtung, Länge, Start- und Endpunkt gespeichert. OKSTRA kennt auch den Lage- und Höhenplan (Gradienten) einer Achse. Das Vokabular des CB-NL stellt hier nur begrenzt Semantiken bereit, um diese Klassen bzw. Attribute zu beschreiben. Die Tabelle in Abbildung 37 vergleicht die Modellierungselemente von OKSTRA und CB-NL in Bezug auf die Trassierungsbeschreibung.

OKSTRA 2.017		CB-NL
Achse		http://ont.cbni.org/cb/def/Middenas
	Bezeichnung	Keine direkte Entsprechung in der CB-NL Mögliches Mapping auf allgemeineres Konzept: http://ont.cbni.org/cb/def/StringProperty
	Art_Achse	Keine direkte Entsprechung in der CB-NL
	Informationstext	Keine direkte Entsprechung in der CB-NL
	Netzfunktion	Keine direkte Entsprechung in der CB-NL
	Strassenkategorie	Keine direkte Entsprechung in der CB-NL
Achselement		Keine direkte Entsprechung in der CB-NL
	Anfangsstation_rechnerisch	Keine direkte Entsprechung in der CB-NL Mögliches Mapping auf allgemeineres Konzept: http://ont.cbni.org/top/def/Geometrischeeigen-schap
	Elementtyp	Keine direkte Entsprechung in der CB-NL
	Laenge	http://ont.cbni.org/top/def/Lengte
	Richtung	http://ont.cbni.org/cb/def/Richtung
	Parameter	Keine direkte Entsprechung in der CB-NL
	Radius_zu_Beginn	Keine direkte Entsprechung in der CB-NL
	Radius_am_Ende	Keine direkte Entsprechung in der CB-NL

Abbildung 37: Vergleich der Modellierungsmöglichkeiten in Bezug auf die Trassierung

2.3 AP 3: Die Grundsätze der Modellierung für eine Übertragung des OKSTRA in OWL

Gegenstand dieses Forschungsvorhabens ist die Untersuchung von Verfahren des „Linked Data“, mit deren Hilfe der Datenaustauschstandard „Objektkatalog für das Straßen- und Verkehrswesen“ (OKSTRA) mit anderen straßenbaurelevanten Schemata verknüpft werden kann. Im Fokus des Projekts liegt dabei der niederländische Katalog CB-NL.

Um das Ziel zu erreichen, den OKSTRA und auf ihm basierende Instanzdatensätze als Linked Data Strukturen darzustellen und zu verarbeiten, sind verschiedene Schritte notwendig:

1. Überführung des Datenschemas (von XMI/XSD in RDF/RDFS/OWL)
2. Überführung von (beispielhaften) Instanzdaten, die momentan als XML oder in Form von WebServices vorliegen in eine der RDF-Serialisierungsformen wie RDF/XML, NTriples, Turtle etc. oder ein Repository/Tripel Store mit entsprechenden Unterstrukturen.
3. Verfügbarmachung beider Datensätze in Linked Data konformen Infrastrukturen (SPARQL Endpoints, Browser Frontend mit HTTP Content Negotiation wie Pubby³² etc.) bzw. zusätzlicher Ebenen von Webservice-ähnlichen Strukturen.

Jeder einzelne dieser drei Hauptbestandteile der Repräsentierung von OKSTRA-Daten als RDF/OWL kann auf unterschiedliche Weise erreicht werden. Obwohl es zahlreiche „Best Practices“ aus anderen Fachbereichen und ähnlich gelagerten Forschungs- und Entwicklungsfragestellungen gibt, hat sich bislang kein „goldener Weg“ etablieren können. Jede Entscheidung in diesen Hauptschritten hat Konsequenzen hinsichtlich der Handhabbarkeit der resultierenden Informationen und Infrastrukturen. Im Folgenden werden diese Konsequenzen, die Vor- und Nachteile der jeweiligen Lösungsansätze erörtert und diskutiert.

In diesem Abschnitt werden die verschiedenen Entscheidungsgrundlagen erörtert, auf deren Basis die möglichen Implementierungen einer Linked Data Repräsentation des OKSTRA und auf ihm basierender Datenmodelle realisiert werden können. Obwohl neben der Verwendung der Ontology Web Language (OWL) weitere Modellierungsvokabulare zur Verfügung stehen, mit deren Hilfe das OKSTRA-Modell mit Linked Data bzw. Semantic Web Prinzipien abgebildet werden können³³, wird für den Kontext dieses Dokumentes die kurze und griffige Bezeichnung „okstraOWL“ verwendet.

In diesem Dokument werden verschiedene Modellierungsoptionen erläutert, die in ihrer Kombination weitreichende Konsequenzen für das Vorhaben haben, OKSTRA und auf ihm basierende Datensätze in der Form von Linked Data zugänglich zu machen. Die jeweiligen Optionen sind thematisch gruppiert und in fett gedruckten **MO.[Kurzname].#** gekennzeichnet. Die Wahl der jeweiligen Ordnungszahl ist dabei unerheblich und dient lediglich zur Identifizierung. Die einzelnen Optionen wurden im Zuge eines Workshops mit den Beteiligten im Februar 2017 diskutiert. Die jeweilig durch die Autoren getroffene Entscheidung ist durch eine entsprechende farbliche Hinterlegung in den Übersichtstabellen gekennzeichnet.

Trotz der Bemühung der Autoren, die nötigsten zugrundeliegenden Prinzipien von Linked Data und Semantic Web kompakt zusammenzufassen, ist die Konsultation von entsprechenden Handbüchern wie z.B. (Allemang and Hendler, 2011) empfehlenswert.

Technische Begriffe werden weitestgehend in ihrem englischen Original gebraucht.

Schlüsselworte, technische Details aus den jeweiligen Schemata und Instanzdaten werden in einem **eigenen Zeichenformat** kenntlich gemacht. Auf die Verwendung von Anführungszeichen wird bei diesen Begriffen und Zitaten durchgehend verzichtet.

Dieser Abschnitt enthält eine kurze Übersicht über die generellen Möglichkeiten der Abbildung der EXPRESS/UML/XMI Modelle in OWL und den zugrundeliegenden Modellierungsprinzipien.

³² <http://wifo5-03.informatik.uni-mannheim.de/pubby/>

³³ Bspw. die Modellierungssprachen RDF Schema (RDFS), SHACL oder SKOS, bzw. die Verschiedenen Profile („Dialekte“) von OWL

2.3.1 Stand der Technik

In Forschung und Entwicklung gibt es zahlreiche Beispiele für die Überführung von Domänenmodellen aus ihren historisch gewachsenen Schemata („Legacy Modellen“) wie NIAM, EXPRESS oder UML in Linked Data Formate und Ontologien. Für ingenieurtechnische Bereiche hat das US-amerikanische „National Institute for Standards and Technology“ (NIST) in der OntoSTEP Initiative bspw. grundsätzliche Vorschläge für die Überführung von EXPRESS Schemata in die Ontology Web Language im „Description Logics“ Profil (OWL-DL) unterbreitet (Krima et al., 2009). Konvertierungen für das Hochbau-Informationsmodell Industry Foundation Classes (IFC) (ISO 16739:2013, 2013) sind in zahlreichen Varianten erstellt und diskutiert worden (Beetz et al., 2009, 2005; Pauwels and Terkaj, 2016; Schevers and Drogemuller, 2005). Von der buildingSMART Organisation wurde im Jahr 2016 nach einem langjährigen Standardisierungsprozess eine erste OWL-basierte Version der Industry Foundation Classes als „ifcOWL“ verabschiedet³⁴.

Die technischen Entscheidungen, die in diesen Initiativen getroffen wurden, basieren auf ausführlichen Auseinandersetzungen mit den jeweiligen Teilaspekten sowohl in den entsprechenden Kommissionen und Arbeitsgruppen als auch auf wissenschaftlichen Plattformen. Die Ergebnisse wurden unter anderem im buildingSMART Modeling and Linking Guide dokumentiert³⁵ und sind in den vorliegenden Bericht eingeflossen.

Grundsätzlich müssen bei Transformationen von Informationsmodellen in Linked Data und Semantic Web Repräsentationen einige grundlegende Rahmenbedingungen beachtet werden, die in den folgenden Abschnitten diskutiert werden.

2.3.2 AAA, OWA und CWA

Ein wichtiges Basisprinzip der Semantic Web (Berners-Lee et al., 2001) und Linked Data (Bizer et al., 2009) Initiativen ist die Erweiterbarkeit von Informationsmodellen und Instanzdatensätzen. Das Prinzip „Anyone can say Anything about Anything“ (AAA)³⁶ ist die fundamentale Ausgangslage, die die Möglichkeiten des World Wide Web (WWW), Texte durch Hyperlinks in vernetzte Hypertexte zu verwandeln auf die Modellierung von anderen, maschinenlesbaren Formen von Wissens-, Informations- und Datenmodellen erweitert. Dem liegt der Gedanke zugrunde, dass Wissen und Information niemals wirklich abgeschlossen und endgültig sind, sondern in einer offenen, vernetzten Umgebung jeweils erweitert werden können und werden. Diese sogenannte „Offene Welt-Annahme“ (Open World Assumption, OWA) unterscheidet sich damit in einigen wesentlichen Punkten von anderen „klassischen“ Informationsmodellen, deren oberste Prinzipien Abgeschlossenheit, Konsistenz und Persistenz sind. Diese „Geschlossenen Welt-Annahme“ (Closed World Assumption, CWA), die auch dem OKSTRA und auf ihm basierenden IT Systemen zugrunde liegt, geht dabei hinsichtlich der Modellierung von stark abweichenden Prinzipien aus.

Als einfaches Beispiel kann beispielsweise die Definition eines Attributes oder einer Relation für eine Klasse/Entity/Konzept herangezogen werden: Unter den Bedingungen der CWA wird für etwa für die Definition der Klasse *Achse* das Attribut *Informationstext* als eine beliebig große Menge mit mindestens einem Element (SET [1: ?]) von Daten des Typs STRING festgelegt. In Datenbanksystemen, Im- und Exportern für domänenspezifischen Anwendungen werden diese Informationen entsprechend implementiert. Die Belegung mit anderen Werttypen (Fließkommazahlen, Objektreferenzen) oder das Fehlen eines Elementes würden von allen Systemen als entsprechende Integritäts- oder Konsistenzfehler behandelt werden und bspw. zum Abbruch des entsprechenden Verarbeitungsprozesses führen.

Unter den Bedingungen der OWA und des AAA jedoch ist ein Datensatz einer *Achse*-Instanz das Attribut *Informationstext* mit einem Wert des Types `xsd:double` zuweist zunächst aus syntaktischer und selbst semantischer Sicht nicht fehlerhaft, da die Aussage „Achse_1 hat Informationstext 1.0f“ in einem anderen Kontext durchaus sinnvoll sein könnte. So könnte selbst das Schema an anderer Stelle um die Möglichkeit

³⁴ <http://www.buildingsmart-tech.org/future/linked-data>

³⁵ [www.modelservers.org/public/bSI/170608_LDWG_Modelling-and-Linking-Guide-\(MLG\).docx](http://www.modelservers.org/public/bSI/170608_LDWG_Modelling-and-Linking-Guide-(MLG).docx)

³⁶ Jeder kann Alles über Alles sagen

der Speicherung eines Fließkommazahlenwertes erweitert worden sein, ohne dass diese Information momentan zugänglich ist. Die Antwort auf die Frage an das System, ob das Datenmodell „korrekt“ sei, heißt im klassischen CWA-Kontext „nein/Fehler“³⁷ während sie unter OWA-Bedingungen lediglich „Kann sein/Es sind nicht alle notwendigen Informationen bekannt“ lautet.

Die Beschränkungen bspw. von Datentypen, Kardinalitäten etc. können jedoch mit entsprechenden Modellierungsvokabularen „nachgerüstet“ werden, um CWA-artige Verhaltensweisen und Operationen in Linked Data Kontexten zu ermöglichen. Mithilfe von OWL etwa lässt sich die formallogische Definition eines Konzeptes „Trasse“ etwa durch die folgenden Axiome aufstellen.

1. „Eine notwendige Bedingung, um ein Ding³⁸ als Element der Menge aller Trasse-Dinge zu klassifizieren, ist das Vorhandensein von mindestens einem Element aus der Menge von Eigenschaften der Eigenschaftsklasse Informationstext.“
2. „Eine weitere notwendige Bedingung ist, dass jedes Element der Menge der Informationstext-Eigenschaften den Datentyp xsd:String hat.“

In OWL kann dies in mit dem folgenden Code (in Turtle-Notation) gefasst werden:

```

:Trasse
  a      owl:Class ;
  rdfs:subClassOf owl:Thing ;
  rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:minCardinality "1"^^xsd:nonNegativeInteger ;
      owl:onProperty :informationstext
    ] ;
  rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:allValuesFrom xsd:string ;
      owl:onProperty :informationstext
    ] .

:informationstext
  a      owl:DatatypeProperty ;
  rdfs:domain :Trasse ;
  rdfs:range xsd:string .

```

Um ein Modell auf seine Konsistenz zu prüfen, können nun generische Inferenzmaschinen (engl. Reasoner) verwendet werden, die den Beweis führen, dass die dem System im Instanzmodell (ABox) bekannten Fakten/Aussagen hinsichtlich der Klassenzugehörigkeiten an die oben (in der TBox) gestellten Randbedingungen erfüllen. Diese Basisprinzipien sind in einer Übersicht in Abbildung 38 dargestellt.

³⁷ Dieses Prinzip wird oft auch als „Negation as Failure“ bezeichnet

³⁸ `owl:Thing` die Superklasse aller Konzepte in der OWL

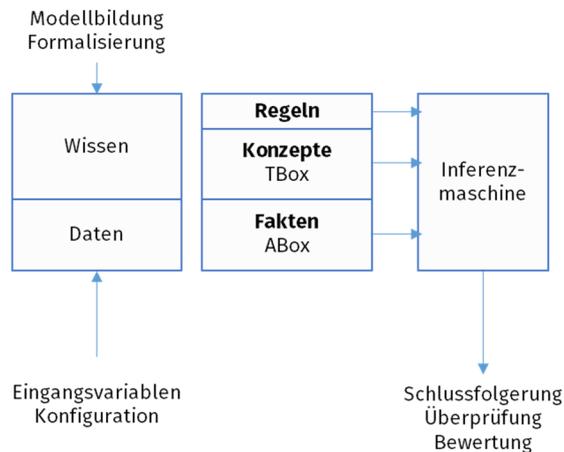


Abbildung 38: Zusammenhang zwischen Konzepten (TBox) und Fakten (ABox)

Diese formallogische Herangehensweise zur Modellierung von Wissen und Fakten basiert einerseits auf gut verstandenen Prinzipien, erfordert jedoch andererseits eine sorgfältige Modellierung und ist mit teils erheblichem Rechenaufwand verbunden: Der Aufwand der Beweisbarkeit/Konsistenzprüfung nimmt mit den verwendeten Konstruktionen/Sprachfeatures der Modellierung von Randbedingungen zu. Auch die Menge der zu betrachtenden Fakten³⁹ trägt zu diesem Aufwand bei. Je nach verwendeter Modellierung können Beweise nicht in garantiert endlicher Zeit ermittelt oder überhaupt entschieden werden. Für eingehende Diskussionen siehe auch Abschnitt 2.3.5.

Ein „preiswerter“ Ausweg aus diesem Dilemma von formaler Logik und Inferenzmaschinen ist die Verwendung von Abfragesprachen wie SPARQL⁴⁰ mit deren Hilfe Modellinkonsistenzen durch Abfragen wie „Ermittle alle Elemente, für die als Klassenzugehörigkeit `Trasse` angegeben wurde, die jedoch **NICHT** mindestens eine `Informationstext` Eigenschaft des Datentyps `xsd:String` im Graphen vorhanden ist“. Solche Abfragen können auch für große Datenmengen performant ausgeführt werden, erfordern aber einen zusätzlichen Aufwand und verlagern Teile der Logik und des Wissens aus dem Modell in externe Systeme. Die Methoden SHACL und SPIN als Modellierungsalternativen zu OWL bieten hier teilweise alternative Lösungswege.

2.3.3 Schemadaten (UML/XMI/EXPRESS -> OWL)

Wie einleitend erläutert, kann der Transformationsprozess grundsätzlich in drei Schritten erfolgen. In diesem Abschnitt wird die Überführung des OKSTRA-Schemas in die Linked Data Entsprechung (TBox) erörtert.

2.3.3.1 URI Komposition

Einer der entscheidenden Faktoren für die Gebrauchsfreundlichkeit von Linked Data Schemata ist die Lesbarkeit und Navigierbarkeit der verwendeten Ressourcen. Dazu gibt es zahlreiche Best Practices aus großen Forschungs- und Entwicklungsinitiativen⁴¹ und aus dem Kontext des W3C⁴². Neben der Wahl der Klassen- und Eigenschaftsnamen selbst, die eingehender in Abschnitt 2.3.3.3 erörtert werden, ist dabei auch die Zurverfügungstellung von einer dereferenzierbaren Version des Schemas entscheidend. Dazu sollten neben dem Gesamtschema oder den modularen Teilschemata in ihrer RDF/XML oder Turtle Repräsentation auch ein SPARQL Endpoint angeboten werden, der mittels „HTTP content negotiation“ entweder den maschinenlesbaren Code (RDF/XML, Turtle) oder die für menschliche Konsumenten vorgehaltene HTML-Dokumentation bereithalten kann.

³⁹ Größe der Datenmodelle

⁴⁰ <https://www.w3.org/TR/sparql11-query/>

⁴¹ https://www.w3.org/2011/gld/wiki/223_Best_Practices_URI_Construction

⁴² <https://www.w3.org/TR/cooluris/>

Die jeweilige Version sollte bereits in der URL erkennbar sein und beispielsweise dem Muster in Listing 1 entsprechen.

```
http://[Domain]/model/[Version]/[Schema]#[Resource]
```

```
http://okstraowl.org/model/2017/okstraowl#Achse
```

Listing 1 Muster und Beispiel für die Namensgebung und URI Komposition

Für den OKSTRA ist im Dokument T0010⁴³ der Umgang mit URL-Namen folgendermaßen festgelegt:

```
http://www.okstra.de/Kontextname/Version/Kontextspezifischer_Pfad
```

2.3.3.2 Datentypen: boxed / unboxed

Bei der Modellierung in RDFS und OWL können grundsätzlich zwei verschiedene Typen von Daten unterschieden werden: Entitäten-/Objektinstanzen und simple Datentypen (Literals) wie Ganzzahl, Zeichenkette, Gleitkommazahl etc. (Integer, String, Float). Sowohl in den ursprünglichen EXPRESS-Schemata als auch in ihren aktuellen XML/XSD Repräsentationen werden diese einfachen Datentypen auch im OKSTRA noch mit zusätzlichen Randbedingung spezifiziert, um ihre intendierte Semantik/Verwendung eindeutig⁴⁴ zu regeln. So wird für die Längenmessung (z.B. `Achselement.Laenge:Meter` in `S_ENTWURF`) meist der Datentyp `Meter` verwendet, der als eine Typdefinition des Elementartypen `REAL` (Express: `TYPE Meter = REAL` in `SCHEMA S_Allgemeine_Objekte`) bzw. `xsd:double` (als Erweiterung des im GML `basicypes.xsd` Schema festgelegten komplexen Datentypen `MeasureType`) modelliert ist. In RDF bzw. RDFS sind zur Erhöhung der Interoperabilität jedoch nur einfache XML Schema Datentypen zugelassen⁴⁵ bzw. von Implementierungen unterstützt. Um den Datentyp `Meter` in einem `okstraOWL` Schema nachzumodellieren, gibt es verschiedene Möglichkeiten:

- **MO.Typen.1** Die Reduzierung auf simple XML Schema Datentypen (hier `xsd:double`) bei Über-
gehung ihres semantisch reicheren Datentyps, also bpsw.

```
okstra:Laenge_Achselement
  a      rdf:Property ;
  rdfs:domain okstra:Achselement ;
  rdfs:range xsd:double .
```

Listing 2 Reduzierung von auf elementare XSD Datentypen **MO.Typen.1**

Bei Verwendung des OWL Vokabulars würde die `rdf:Property` durch die entsprechenden `owl:DatatypeProperty` ersetzt. In dieser Variante taucht der erweiterte Datentyp `Meter` aus dem `OKSTRA` Schema nicht mehr auf.

- **MO.Typen.2** Die Definition einer kapselnden Objektklasse, die den Namen des originalen `OKSTRA` Datentypes (hier: `Meter`) erhält und den Subtyp mit der adäquaten `xsd:Datentype` vorhält. Das klassenspezifische Attribut (hier `Laenge`) wird in diesem Fall als `owl:ObjectProperty` (und nicht als `owl:DatatypeProperty`) modelliert. Die Basistype `Meter` wird ebenfalls als Klasse modelliert, die wiederum Subklasse des entsprechenden Basistypen (hier `Double` oder wie in `ifcOWL` `expr:REAL` ist. Dieser Basistyp wiederum hat eine `owl:DatatypeProperty hasDouble` die einen zulässigen Wer-

⁴³ <http://www.okstra.de/docs/t-dokumente/t0010.pdf>

⁴⁴ Oder zumindest weniger mehrdeutig

⁴⁵ Siehe „XML Schema Datatypes in RDF and OWL“ <https://www.w3.org/TR/swbp-xsch-datatypes/> Sowie „RDF Semantics“ <https://www.w3.org/TR/2004/REC-rdf-mt-20040210/#DTYPEINTERP> für die Version RDF 1.0 bzw. <https://www.w3.org/TR/rdf11-concepts/#dfn-recognized-datatype-iris> für die RDF Version 1.1 die alle eingebauten XSD Datentypen identifiziert: <https://www.w3.org/TR/rdf11-concepts/#xsd-datatypes>

tebereich des entsprechenden XSD-Datentypen hat. Dieses Modellerschema von Datentypen als Objektklassen ist in Programmiersprachen wie Java oder C# als „Boxing“ oder „Wrapping“ wiederzufinden.

```

okstra:Double
  rdf:type owl:Class ;
  rdfs:subClassOf
    [ rdf:type owl:Restriction ;
      owl:allValuesFrom xsd:double ;
      owl:onProperty :hasDouble
    ] .

okstra:Laenge_Achselement
  rdf:type owl:ObjectProperty ;
  rdfs:domain okstra:Achselement ;
  rdfs:range okstra:Meter .

okstra:Meter
  rdf:type owl:Class ;
  rdfs:subClassOf :Double .

okstra:hasDouble
  rdf:type owl:DatatypeProperty ;
  rdfs:domain okstra:Double ;
  rdfs:range xsd:double .

```

Listing 3 Gekapselte komplexe Datentypen der **MO.Typen.2**

Erwägungen:

	Vorteile	Nachteile
MO.Typen.1: Einfache Datentypen	<ul style="list-style-type: none"> - Kompaktes Schema - Kompakte Instanzdaten 	<ul style="list-style-type: none"> - Verlust von Semantik durch fehlende Typenbezeichner
MO.Typen.2: Komplexe Datentypen (boxed)	<ul style="list-style-type: none"> - Vollständiger Erhalt der zugrunde liegenden Semantik 	<ul style="list-style-type: none"> - Komplexes Schema - Komplexe Instanzdatengraphen - Mehrere Tripel-Aussagen für einfache Werte nötig

2.3.3.3 Klassen und Namensgebung

Die Überführung der Klassenstruktur des OKSTRA inklusive der Mehrfachvererbung lässt sich durch einfache Transformationen von `<element xmi:type="uml:Class">` im XML Schema bzw. ENTITY in Express in `owl:Class` Definitionen erreichen.

Im Falle von OWL müssen alle Klassen zusätzlich als „disjunkt“ (engl. disjoint) deklariert werden, d.h. dass eine Objektinstanz nicht gleichzeitig Teil zweier Klassenzugehörigkeitsmengen sein kann.

Eine kritische Frage der Modellierung großer Modelle wie dem OKSTRA ist die Namensgebung der Klassen. Obwohl die Klassen innerhalb des OKSTRA jeweils nur einmal vorkommen gilt im SW/LD Kontext nicht die Unique Name Assumption (UNA) die besagt, dass zwei verschiedenen Namen auch zwei verschiedenen Konzepte/Klassen bezeichnen.

Grundsätzlich könnten Klassen alternativ mit ihren jeweiligen ENTITY Namen oder mit den UUIDs aus dem derzeitigen XMI-Modell versehen werden, wobei der Name der Klasse als eine `rdfs:label` an die `owl:Class` gehangen wird.

```
<packagedElement xmi:type="uml:Package" xmi:id="EAPK_C58C9403_1B89_41cd_94CF_68386ACF6FE3"
name="S_Entwurf" visibility="public">
  <packagedElement xmi:type="uml:Class"
xmi:id="EAID_E4106DA8_B309_4197_A6A5_37570759D8B5" name="Achse"
visibility="public">
  </packagedElement>
</packagedElement>
```

Listing 4 Verkürzte Darstellung der XML Definition der Achse Klasse

```
okstra:EAID_E4106DA8_B309_4197_A6A5_37570759D8B5
  rdf:type owl:Class ;
  rdfs:label "Achse"^^xsd:string@de ;
  rdfs:label "Axis"^^xsd:string@en .
```

Listing 5 **MO.Namen.1** Beispiel der TTL Wiedergabe mit UUIDs als Klassennamen

```
okstra:Achse
  rdf:type owl:Class ;
  rdfs:label "Achse"^^xsd:string@de ;
  rdfs:label "Axis"^^xsd:string@en ;
  xmi:id "EAID_E4106DA8_B309_4197_A6A5_37570759D8B5"^^xsd:string .
```

Listing 6 **MO.Namen.2** Beispiel der TTL Notation mit (deutschen) Namen als Klassennamen sowie einer eigenen Eigenschaft zum Erhalt der UUID

	Vorteile	Nachteile
MO.Namen.1: UUIDs	<ul style="list-style-type: none"> - Eindeutige, sprachunabhängige Definition - Vereinfachung von Roundtripping Modellierwerkzeug - Vereinfachte Internationalisierung 	<ul style="list-style-type: none"> - Schwer lesbar - Umständliche Navigation - Lange Zeichenketten führen zu großen Dateien
MO.Namen.2: XMI/XSD/EXPRESS „natürliche“ Klassennamen	<ul style="list-style-type: none"> - Gut lesbar für deutschsprachige Fachleute - Kompakt - Mehrsprachigkeit durch zusätzliche Labels realisierbar 	<ul style="list-style-type: none"> - Stellenweise Kollisionen durch Mehrfachverwendung - Semantik im Namen - Internationalisierung erfordert redundantes Schema und <code>owl:sameAs</code> / <code>owl:EquivalentClass</code> Mappings

2.3.3.4 Attribute und Relationen `owl:DatatypeProperty` / `owl:ObjectProperty`

Für eine grundsätzliche Diskussion der Fragen der Abbildung von simplen Datentypen von Attributen aus dem OKSTRA in okstraOWL in gekapselter oder direkter Form wird auf Abschnitt 2.3.3.2 verwiesen.

Im Unterschied zu den „lokalen“ Definitionen von Attributen in EXPRESS oder auch XML und XML Schema, sind die Definitionen von Eigenschaften und Attributen in der Form von `rdf:Property` und ihrer abgeleiteten Formen `owl:DatatypeProperty` und `owl:ObjectProperty` immer „global“. Dieser Umstand hat Konsequenzen: Das Attribut `Laenge` wird im OKSTRA in den jeweiligen lokalen Kontexten unterschiedlicher Klassen mit unterschiedlichen Wertebereichen / Typen / Maßeinheiten belegt. So hat die Klasse `Achselement` im Schema `S_Entwurf` ein Attribut `Laenge` vom Typ `Meter`, während im Schema `S_Strassennetz` die Klasse `Abschnitt_oder_Ast` ein Attribut desselben Namens mit einem anderen Typen (`Kilometer`) definiert. Eine global definierte `rdf:Property` könnte lediglich indirekt über `owl:Restriction` Einschränkungen auf den jeweils unterschiedlichen Wertetyp festgelegt werden.

```

ENTITY Achselement
SUBTYPE OF (OKSTRA_Objekt);
--- Attribute :
  Laenge          : Meter;
END_ENTITY;

```

Listing 7 Verwendung eines Attributes `Laenge` des Types `Meter` in der verkürzten Definition der Klasse `Achselement`

```

ENTITY Abschnitt_oder_Ast
--- Attribute :
  Laenge          : Kilometer;
END_ENTITY;

```

Listing 8 Verwendung eines Attributes `Laenge` des Types `Kilometer` in der verkürzten Definition der Klasse `Abschnitt_oder_Ast`

Ein gängiger Ausweg liegt darin, die Attribute auch in RDF zu „lokalisieren“, indem eine jeweils eigene Eigenschaft für die entsprechende Klasse definiert wird, etwa `Laenge_Achselement` und `Laenge_Abschnitt_oder_Ast`, denen dann über einfachere RDFS `domain/range` Beschränkungen der entsprechende Basistyp zugewiesen werden kann.

2.3.3.5 Schlüsseltabellen: Enumerationen, eigene Klassen, Unterklassen

Eine Besonderheit des OKSTRA-Modells ist das Konzept der Schlüsseltabellen, die aufgrund ihrer (historischen) Designentscheidungen besonderer Aufmerksamkeit gegenüber anderen Modellkonvertierungen (etwa ifcOWL) bedürfen. Diese Besonderheiten sind bereits in der Umstellung von EXPRESS auf XML-Schema mit entsprechenden Konstruktionen versehen worden, die in Abschnitt 4.9 des Dokumentes „T0006 - Ableitung OKSTRA®-XML aus der EXPRESS-Modellierung“⁴⁶ diskutiert werden. Drei verschiedenen Darstellungsvarianten werden in diesem Dokument spezifiziert: Verweis, Einbettung und Kompaktdarstellung. Alle Varianten sind in den zur Verfügung stehenden Testdatensätzen anzutreffen.

Im Prinzip handelt es sich bei den Schlüsseltabellen um weitere Auszeichnungen von Objektklassen, wie sie ähnlich auch durch Subklassen modelliert werden könnten. Stattdessen sind diese Auszeichnungen in Schlüssel-Werte-Paar-Tabellen abgelegt. Dies hat einerseits den Vorteil, leichter dynamische Erweiterungen realisieren zu können, ohne die Gesamtzahl der vorhandenen Klassen zu erhöhen. Andererseits ist die Semantik dadurch „indirekt“ und nicht mit andernorts gebräuchlichen Ansätzen (z.B. Enumerationen) abzubilden.

```

<packagedElement xmi:type="uml:Class" xmi:id="EAID_E4106DA8_B309_4197_A6A5_37570759D8B5"
name="Achse" visibility="public">
  <ownedAttribute xmi:type="uml:Property"
xmi:id="EAID_A97795F7_6149_460d_9033_B3A6CD5E32B8" name="Art_Achse"
visibility="public" isStatic="false" isReadOnly="false"
isDerived="false" isOrdered="false" isUnique="true"
isDerivedUnion="false">
  </ownedAttribute>
</packagedElement>

<element xmi:idref="EAID_DAEA3114_A1A9_4c38_B748_CF1710A3FC58"
xmi:type="uml:Class" name="Art_Achse" scope="public">
  <properties documentation="Schlüsseltabelle zur Angabe der Art einer
  &lt;i&gt;Achse&lt;/i&gt;" isSpecification="false" sType="Class"
  nType="0" scope="public" stereotype="Schluesseltabelle" isRoot="false"
  isLeaf="false" isAbstract="false" isActive="false"/>
  <constraints>
    <constraint name="Werteliste" description="&#xA;'1', 'Hauptachse'&#xA;'2', 'Gra-
bensohle'&#xA;'3', 'Radweg'&#xA;'4', 'Dreiecksinsel'&#xA;'5', 'Tropfen'&#xA;'6',
'Trassenbegleitende Bauwerke'&#xA;'7', 'Randausrundung'&#xA;'8',

```

⁴⁶ <http://www.okstra.de/docs/t-dokumente/t0006.pdf>

```
'Fahrbahnrand'&#xA;'9',      'ASB-Bestandsachse'&#xA;'10',
'ASB-Hilfsachse'&#xA;'11',  'Nebenachse'&#xA;'12',      'Gehweg'&#xA;'13',
'Gleisachse'&#xA;'14',     'Gebäudeachse'&#xA;'15',    'Bauwerksachse'&#xA;'99',
'Sonstiges'" type="Invariant" weight="0,00" status="Proposed"/>
</constraints>
```

Listing 9 Verkürzte Definition der Klasse Achse mit dem Attribut Art_Achse, dessen Werte aus der namensgleichen Schlüsselstabelle (Definition ebenfalls verkürzt) gewählt werden kann.

Drei grundsätzliche Herangehensweisen für eine Konvertierung in okstraOWL können unterschieden werden:

- a) **MO.Schlüssel.1** Abbildung als Unterklassen der zugehörigen Basisklasse
Mit dieser Option lassen sich Instanzen der jeweiligen Subklasse generieren.

```
okstra:Achse
  rdf:type owl:Class .

okstra:Art_Achse
  rdf:type owl:Class .

okstra:Hauptachse
  rdf:type owl:Class ;
  rdfs:subClassOf okstra:Art_Achse ;
  okstra:Langtext "Hauptachse"^^xsd:string .
  okstra:Kennung "1"^^xsd:string .

okstra:Art_Achse_Achse
  rdf:type owl:ObjectProperty ;
  rdfs:domain okstra:Achse ;
  rdfs:range okstra:Art_Achse .
```

Listing 10 Schlüsselstabelle als Unterklassen in der MO.Schlüssel.1

- b) **MO.Schlüssel.2** Abbildung als Enumeration / `owl:ObjectOneOf`

```
okstra:Achse
  rdf:type owl:Class .

okstra:Art_Achse
  a owl:Class ;
  rdfs:subClassOf expr:ENUMERATION ;
  owl:ObjectOneOf [
    okstra:Hauptachse
  ] .

okstra:Hauptachse
  a Art_Achse, owl:NamedIndividual ;
  okstra:Kennung "1"^^xsd:string ;
  okstra:Langtext "Hauptachse"^^xsd:string .

okstra:Art_Achse_Achse
  rdf:type owl:ObjectProperty, owl:FunctionalProperty ;
  rdfs:domain okstra:Achse ;
  rdfs:range okstra:Art_Achse .
```

Listing 11 Schlüsselstabelle als Enumeration in der MO.Schlüssel.2

c) **MO.Schlüssel.3** Kennzeichnung durch String-Labels

```

okstra:Art_Achse_Kennung_Achse
  rdf:type owl:DatatypeProperty;
  rdfs:domain okstra:Achse ;
  rdfs:range xsd:string .

okstra:Art_Achse_Langtext_Achse
  rdf:type owl:DatatypeProperty;
  rdfs:domain okstra:Achse ;
  rdfs:range xsd:string .

```

Listing 12 Abbildung von Schlüssel Tabellen mithilfe einfacher String in **MO.Schlüssel.3**

Diese Variante erlaubt es auch, die diejenigen Fälle zu berücksichtigen, in denen Schlüssel Tabellen eine Vielzahl von Kennungen umfassen.

	Vorteile	Nachteile
MO.Schlüssel.1	<ul style="list-style-type: none"> - Konventionell, leicht verständliche Abbildung 	<ul style="list-style-type: none"> - Weiter Verästelung des Klassenbaums - Keine echten Diskriminatoren - Kennung oder Langtext müssen parallel (<code>owl:sameAs</code>) vorgehalten werden oder über andere Mechanismen eingebettet werden.
MO.Schlüssel.2	<ul style="list-style-type: none"> - Beibehaltung des Klassenbaums aus EXPRESS/XSD/XMI - Kompatibel zu Ansätzen aus ifcOWL 	<ul style="list-style-type: none"> - Abbildung von Kennung und Langtext nur umständlich erreichbar
MO.Schlüssel.3	<ul style="list-style-type: none"> - Einfache, platzsparende Variante - Schlüssel und Kennung könnten als Mikroformat ähnlich UML/XMI vorgehalten werden (CSV, JSON) - Ähnlich dem original OKSTRA 	<ul style="list-style-type: none"> - Übernahme der schwächeren Semantik „versteckt“ in Strings

Tabelle 1 Übersicht über Vor- und Nachteile von Modellierungsoptionen hinsichtlich der Abbildung von Schlüssel Tabellen

2.3.3.6 Aggregierte Daten / Kollektionen

Aggregierte Datentypen wie `ARRAY`, `SET` und `LIST`, und hier insbesondere die geordneten Varianten, sind nicht eins-zu-eins in OWL übersetzbar. Grund hierfür ist, dass RDF und OWL auf Mengen von Elementen basiert sind. Geordnete Datentypen sind nur umständlich in RDF abzubilden und brauchen eine zusätzliche Schicht um kompatibel mit OWL-unterstützenden Inferenz-Maschinen zu sein (Drummond et al., 2006). Um diesem Umstand Rechnung zu tragen, können in okstraOWL drei Optionen betrachtet werden.

- **MO.Listen.1** Verwendung von `rdf:list`, Bruch mit OWL DL⁴⁷ Kompatibilität. In dieser Option werden schemaseitig keine weiteren Festlegungen gemacht. Die Ordnung einzelner Instanzdaten wird der Implementierung überlassen. Dabei werden die Listenelemente (z.B. Achselemente, siehe Listing 13) zunächst als ungeordnete Relationen an die entsprechende Klasseninstanz gehangen. Anschließend können diese mit herkömmlichen `rdf:list` Elementen in die entsprechende Ordnung gebracht werden. Werden diese Listen in einen externen Graphen ausgelagert, der nur im Bedarfsfall eingebunden wird, bleibt der Instanzgraph kompatibel mit OWL DL.
- **MO.Listen.2** Adaptierung des `expr:LIST` und ifcOWL Ansatzes mit eignen Listentypen pro Klasse und Attribut. In ifcOWL wird eine OWL DL kompatible Nachbildung des `rdf:list` Mechanismus ver-

⁴⁷ OWL Description Logic: Variante der OWL, die Beschreibungslogik formal umsetzt

wendet, der auf die Arbeit von (Drummond et al., 2006) zurückgeht. Dafür werden für alle geordneten Kollektionen eigene Klassen angelegt. Äquivalent sähe dies für die okstraOWL anhand des Beispiels der Relation `Achse.hat_Achselement` des Schemas `S_Entwurf` wie folgt aus:

```
ENTITY Achse
SUBTYPE OF (Status_Eigenschaft);
  hat_Achselement      : LIST [1:?] OF Achselement;
END_ENTITY;
```

Listing 13 Verkürzte EXPRESS Definition der `Achse` Klasse im Schema `S_Entwurf`

```
okstra:Achselement_List a owl:Class ;
  rdfs:subClassOf list:OWList ;
  rdfs:subClassOf [ a owl:Restriction ;
                    owl:allValuesFrom okstra:Achselement_List ;
                    owl:onProperty list:hasNext
                  ] ;
  rdfs:subClassOf [ a owl:Restriction ;
                    owl:allValuesFrom okstra:Achselement_List ;
                    owl:onProperty list:isFollowedBy
                  ] ;
  rdfs:subClassOf [ a owl:Restriction ;
                    owl:allValuesFrom okstra:Achselement ;
                    owl:onProperty list:hasContents
                  ] .

okstra:hat_Achselement_Achse
  a owl:ObjectProperty , owl:FunctionalProperty ;
  rdfs:domain okstra:Achse ;
  rdfs:label "hat_Achselement" ;
  rdfs:range ifc:Achselement_List .

okstra:Achse a owl:Class ;
  rdfs:subClassOf [ a owl:Restriction ;
                    owl:onClass okstra:Achselement_List ;
                    owl:onProperty okstra:hat_Achselement_Achse;
                    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
                  ] ;
  rdfs:subClassOf [ a owl:Restriction ;
                    owl:allValuesFrom [ a owl:Restriction ;
                                          owl:onProperty list:hasNext ;
                                          owl:someValuesFrom [ a owl:Restriction ;
                                                                owl:onProperty list:hasNext;
                                                                owl:someValuesFrom okstra:hat_Achselement_Achse
                                                                ]
                                        ] ;
                    owl:onProperty okstra:hat_Achselement_Achse
                  ] ;
  rdfs:subClassOf [ a owl:Restriction ;
                    owl:allValuesFrom okstra:Achselement_List ;
                    owl:onProperty okstra:hat_Achselement_Achse
                  ] ;
```

Listing 14 Verkürzte Darstellung einer OWL DL kompatiblen geordneten Kollektion von Objekten, hier `hat_Achselement` der Klasse `Achse`

- **MO.Listen.3** Geordnete Kollektionen durch Annotationen (proprietäre eigene Lösung)

	Vorteile	Nachteile
MO.Listen.1 rdf:List	- Standardlösung mit Unterstützung durch Tooling	- Bruch mit OWL DL Kompatibilität - Aufgebläht - Tiefe Graphen
MO.Listen.2 Drummond et al	- OWL DL kompatibel - Kompatibel mit Modellierung ifcOWL	- Umständlich und ausschweifend - Tiefe Graphen - Erschwerte Abfragen (SPARQL)
MO.Listen.3 Annotationen	- Kompakter und platzsparender als andere Optionen	- Semantisch schwach - Fehleranfällig durch Stringbasierung

2.3.3.7 Kardinalitäten

- **MO.Kard.1** Weglassen von Kardinalitätsbeschränkungen. In dieser Option werden die Kardinalitäten, z.B. für die minimale Anzahl von Achselementen einer Achse aus dem Kernschema weggelassen. Der Vorteil ist ein schlankeres Schema. Die Semantik und Überprüfbarkeit der Integrität eines gegebenen Datensatzes nehmen dafür jedoch ab.
- **MO.Kard.2** Optionale Hinzufügung (in externem Schema) als DL-kompatible Lösung ähnlich ifcOWL. Wie im Listing 14 dargestellt, werden hier die minimalen/maximalen Kardinalitäten als owl:Restrictions für die jeweiligen Attribute/Relationen modelliert. Eine Möglichkeit, das Kernschema schlank und übersichtlich zu halten, liegt darin, diese Einschränkungen in einem weiteren (Zusatzschema) zu definieren.
- **MO.Kard.3** Integration in Schema

2.3.3.8 Integration GML

Seit der Umstellung von EXPRESS auf UML/XML in der OKSTKRA Version 2.015, sind die eigenen Basis-Geometrielemente durch ISO 19107:2003 Spatial Schema ersetzt. Damit werden die vom Open Geospatial Consortium (OGC) in der Geographic Markup Language (GML) festgelegten Mechanismen zur Darstellung von Geometrie verwendet. Da das GML-Format seit Anbeginn der Initiative auf dem Resource Description Framework (RDF) aufgebaut wurde, ist eine Migration in Richtung Linked Data ein naheliegender Schritt. Die im folgenden Punkt MO.5.1 skizzierte Lösung ist lediglich für eventuelle Abwärtskompatibilität mit historischen Datensätzen in Betracht zu ziehen.

- **MO.GML.1** Einheitliche Modellierung aus Express-Schema. In dieser (nicht zu empfehlenden) Herangehensweisen werden die in Version 1.015 letztmalig verwendeten Klassendefinitionen wie in Listing 15 ähnlich ifcOWL mit eigenen Listen Elementen (siehe oben) umgesetzt. Die zu erwartenden Graphengrößen und –tiefen lassend diese Herangehensweise nicht wünschenswert erscheinen.

```

ENTITY Linie
--- Attribute :
--- Relationen :
  besteht_aus_Linienelementen : LIST [1:?] OF Linienelement_3D;
  beschreibt_Linien_auf_Flelem : OPTIONAL SET [1:?] OF Linie_Flaechenelement;
END_ENTITY;

ENTITY Linienelement_3D
SUBTYPE OF (OKSTRA_Objekt);
--- Attribute :
  Unterbrechung : OPTIONAL BOOLEAN;
--- Relationen :
```

```

beginnt_bei_Punkt      : Punkt;
endet_bei_Punkt        : Punkt;
: Linie FOR besteht_aus_Linienelementen;
END_ENTITY;

```

Listing 15 Verkürzte Klassendefinitionen von Geometrien in Version 1.015

- **MO.GML.2.** Übernahme GML/GeoSPARQL Ansätze mit WKT. In dieser Variante werden die vom OGC entwickelten Ansätze zur platzsparenden, effizienten Darstellung von Geometrien als RDF Graphen mittels des sog. „Well-Known Text“ (WKT) übernommen. Da insbesondere Listen, wie sie etwa für Linien, Polygone etc. unerlässlich sind, nur äußerst umständlich in RDF mit den entsprechenden `rdf:list` Elementen dargestellt werden können, ist die Basisidee von WKT, Punkte, Linien, Polygone etc. als Literal-Werte, d.h. Zeichenketten zu erfassen. Ein Beispiel aus dem OGC Dokument 11-052r4 “OGC GeoSPARQL - A Geographic Query Language for RDF Data”⁴⁸ in Listing 16 zeigt anhand eines Liniensegmentes die prinzipielle Vorgehensweise. In ifcOWL, in deren erster Version die EXPRESS-basierten Listen von Einzelkoordinaten (siehe auch 2.3.3.6) eingesetzt wurden, haben erste Experimente auf eine Umstellung in das kompaktere WKT Format zu erheblichen Effektivitätssteigerungen hinsichtlich Graphen- und Dateigrößen geführt (Pauwels et al., 2015).

```

<sf:LineString rdf:about= "http://example.org/ApplicationSchema#EExactGeom">
  <geo:asWKT rdf:datatype= "http://www.opengis.net/ont/geosparql#wktLiteral">
    <![CDATA[
      <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
      LineString((-83.4 34.0, -83.3 34.3))
    ]]>
  </geo:asWKT>
</sf:LineString>

```

Listing 16 Beispiel einer LineString Geometrie als WKTLiteral in einem RDF/XML Graphen

2.3.4 Instanzdaten

Dem konzeptionellen Ansatz von Linked Data entsprechend, sollten Instanzdaten von okstraOWL vor allem hinsichtlich ihrer Handhabbarkeit in verteilten, netzwerkbasierten Umgebungen betrachtet werden. Anders als bei der klassischen Datei-basierten EXPRESS- und XML-Datenverwaltung stellt das Prinzip einer Resource, die über eine entsprechende URI identifizierbar und erreichbar ist, die Grundlage dar. Auf Schema- und Instanzebenen stellt das Versenden etwa von Graphen als RDF-Dateien nur eine der möglichen Verarbeitungsformen dar.

2.3.4.1 Struktur und Modularisierung

Die Strukturierung und Modularisierung von Schema- und Instanzdaten muss flexibel und granular auf die jeweiligen Einsatzzwecke abgestimmt werden. Ähnlich dem Konzept der Profile⁴⁹ in OKSTRA kann auch okstraOWL als einsatz- und kontextgebundene Teilschemata statt des monolithischen Gesamtschemas gehandhabt werden. Ebenso wie die TBox (das Schema) können auch Instanzdaten (die ABox) entlang verschiedener virtueller Grenzen aufgeteilt werden.

2.3.4.2 Modularisierung Schema

- **MO.Mod.1** Komplettschema. In dieser Option wird das Gesamtmodell aus der XMI-Datei mit allen Modellierungsaspekten in einem einzigen Langschema zusammengefasst.
- **MO.Mod.2** Modularisiertes Schema mit Kern. In dieser Option werden alle Klassendefinitionen und Vererbungsinformationen zusammen mit den Eigenschaften (`owl:DatatypeProperty` und `owl:ObjectProperty`) mit `domain/range` Angaben in ein Kernschema geschrieben, alle `owl:Restrictions` jedoch in einem separaten Schema vorgehalten (z.B. `okstraOWL-restrictions.rdf`).

⁴⁸ https://portal.opengeospatial.org/files/?artifact_id=47664

⁴⁹ Festlegung von inhaltlichen Ausschnitten (Profilen) des OKSTRA-Datenmodells, <http://www.okstra.de/docs/t-dokumentation/t0009.pdf>

Einer vorsichtigen Schätzung zufolge würde dies eine Verminderung der Anzahl der Tripels des Schemas um ca. 35% bedeuten.

- **MO.Mod.3.** Modularisiertes Schemata entlang der EXPRESS und XSD-Teilschemata. Um die Implementierung des Standards für solche Anwendungsszenarien einfacher zu machen, die nur einen Teil des im OKSTRA Modells benötigen, lassen sich analog zu den EXPRESS-Teilschemata und den XML packages auch in okstraOWL modulare Teilschemata bilden. Die Einbindung der externen Abhängigkeiten (S_Geometrieschema von S_Entwurf) erfolgt durch Referenzierung
- **MO.Mod.4** Kombinationen MO.5.2/MO.5.3.

	Vorteile	Nachteile
MO.Mod.1: Komplettschema	<ul style="list-style-type: none"> - Übersichtlich - Kompakt 	<ul style="list-style-type: none"> - Größeres Modell - Weiterführende OWL Modellkonstruktionen von manchen Werkzeugen nicht unterstützt
MO.Mod.2: Modularisiertes Schema mit Kern	<ul style="list-style-type: none"> - Getrennte Modellierungsfeatures - Kleinere Teilschemata 	<ul style="list-style-type: none"> - Fragmentierung
MO.Mod.3: Schemata entlang der EXPRESS und UML-Teilschemata	<ul style="list-style-type: none"> - Bessere Wiederverwendbarkeit durch verringerte Komplexität („separation of concerns“) - Kleine Teilschemata 	<ul style="list-style-type: none"> - Fragmentierung
MO.Mod.4:	<ul style="list-style-type: none"> - Flexible Modularisierung 	<ul style="list-style-type: none"> - Fragmentierung

2.3.4.3 Tripels/Quadruples

In Anwendungsszenarien, in denen beispielsweise die Historisierung von Objekten unterstützt werden soll, kann es vorteilhaft sein, die Konzepte von Kontexten und Named Graphs in RDF auszunutzen. Dabei wird den herkömmlichen RDF 3-Tupeln jeweils noch ein viertes Element zugewiesen, dass den Graphen erfasst und damit einen Kontext herstellt. Mit diesem Mechanismus lassen sich etwa verschiedene Aspektmodelle parallel zueinander vorhalten.

2.3.4.4 Serialisierung

Die für RDF zur Verfügung stehenden Serialisierungsformate wie RDF/XML⁵⁰, TURTLE⁵¹, N-Triples⁵², Notation3 N3⁵³ etc. sind prinzipiell gleichwertig, eignen sich aber unterschiedlich gut für verschiedene Anwendungszwecke:

- Wo eine erhöhte Lesbarkeit erwünscht ist, sollten TURTLE (.ttl) oder N3 Notationen der Vorrang gegeben werden. Das okstraOWL Schema sollte neben RDF/XML auch als TURTLE offiziell veröffentlicht werden.
- Wo schneller Import und Export nötig sind, sind N-Triples oder N-Quads Notationen empfehlenswert, da weder speicherintensiv das komplette XML Domain Object Models (DOM) aufgebaut noch Prefixes aufgelöst werden müssen. Die Wiederholungen der URIs lassen in Klartextnotationen die Dateigrößen erheblich anwachsen, können aber effektiv komprimiert und damit gut transportiert und archiviert werden.

⁵⁰ <https://www.w3.org/TR/rdf-syntax-grammar/>

⁵¹ <https://www.w3.org/TeamSubmission/turtle/>

⁵² <https://www.w3.org/2001/sw/RDFCore/ntriples/>

⁵³ <https://www.w3.org/TeamSubmission/n3/>

- XML/RDF kann durch seine XML-Struktur auch von Nicht-RDF-Werkzeugen verarbeitet werden und ist breit unterstützt. Deshalb ist es empfehlenswert das Schema (parallel zu Turtle) auch in diesem Format vorzuhalten.

2.3.4.5 Tripel/Quadstore Architekturen

Schemaseitig sollten alle gegenwärtigen und zukünftigen Versionen des Schemas in separaten „Named Graphs“ vorgehalten werden, die die langfristige Erreichbarkeit auch nicht mehr aktueller Versionen erlaubt.

Die in Abschnitt 2.3.3.1 dargelegten Erwägungen für die URI-Vergabe (Versionsnummer etc.) sowie der möglichen Modularisierung des Schemas (siehe Abschnitt 2.3.4.2) können mithilfe dieser Strukturen umgesetzt werden.

Auf der Ebene der Instanzdaten müssen mögliche Systemarchitekturen auf den jeweiligen Anwendungsfall abgestimmt werden. Besondere Berücksichtigung sollten dabei datenschutzrelevante Aspekte eventuell sensibler Daten finden. Hier stehen aufgrund der historischen Tradition der möglichst ungehinderten Zugänglichmachung von Daten in der Semantic Web und Linked Data Initiative momentan gegenüber herkömmlichen datenbankbasierten Architekturen noch weniger Möglichkeiten zur Verfügung. Die Verwendung von Systemen, die die Linked Data Platform⁵⁴ (LDP) Standards unterstützen, sind zu empfehlen.

2.3.5 Auswirkungen hinsichtlich der Handhabbarkeit, rechnerischen Komplexität und Beweisbarkeit

Die in den einzelnen Abschnitten diskutierten Modellierungsoptionen haben Auswirkungen auf alle Aspekte des Umgangs mit OKSTRA-Daten in der Form von Linked Data. In den folgenden Abschnitten werden diese Auswirkungen auf bestimmte Aspekte zusammengefasst. Eine kompakte tabellarische Übersicht über die Auswirkungen einzelner Modellierungsoptionen in bestimmten Kategorien finden sich im LDAC 2015 Abschlussreport⁵⁵.

Die unterschiedlichen Aspekte der Modellierungsoptionen können grob in folgende Aspekte unterteilt werden:

- **Datengrößen (Anzahl Tripels).** Die Größe von Datenbeständen sind vor allem bei der Handhabung von Instanzdaten von entscheidender Bedeutung für den effizienten Gebrauch. In RDF Graphen ist dabei die Anzahl der Tupel die entscheidende Größe, da Serialisierungen in verschiedenen Formaten bei gleichem Inhalt stark im benötigten Speicherplatz schwanken können (siehe auch Abschnitt 2.3.4.4). Dabei gilt generell, dass jedes weitere Sprachfeature wie Listen zusätzliche Tupel erfordern und damit die Größe des Graphen erhöhen.
- **Graph-Tiefen.** Je mehr Modellierungsmöglichkeiten der verschiedenen OWL-Profilen bzw. anderer Sprachen und Metamodelle verwendet werden, desto komplexer und verzweigter werden die resultierenden Schema- und Instanzgraphen. Dies hat vor allem negative Auswirkungen auf Abfragen (siehe SPARQL)
- **SPARQL-Queries.** Unmittelbar betroffen von den Tiefen der resultierenden Graphen ist die Zugänglichkeit von Informationen durch standardisierte Abfragesprachen und Regelsprachen wie RuleML, SWRL, N3 etc. Datentypenkapselung, die Verwendung von eigenen, DL-kompatiblen Listentypen oder die Verwendung verschiedener Module z.B. in Named Graphs erhöhen die Komplexität von Abfragen erheblich.
- **Reasoning-Möglichkeiten.** Schon geringe Abweichung formallogischer Modellierung und Stringenz in Schema- und Instanzmodellen sowie die Verwendung nicht-kompatibler Modellierungskonstrukte können den Gebrauch von Inferenzmaschinen erheblich erschweren oder gänzlich unmöglich machen.

⁵⁴ <https://www.w3.org/TR/ldp/> und <https://www.w3.org/TR/ldp-bp/>

⁵⁵ http://ldac-2015.bwk.tue.nl/LDAC_2015_workshopreport.pdf

- **Internationalisierung / Übersetzbarkeit in andere Sprachen.** Bei einer direkten Transformation des bestehenden Modells kann durch den Gebrauch von mehrsprachigen `rdfs:label` Auszeichnungen bereits mit bestehenden Modellierungswerkzeugen⁵⁶ eine Unterstützung der Internationalisierung erreicht werden. Diese zusätzliche Auszeichnungen decken auch Beziehungen zwischen Elementen (`rdf:Property` und ihrer Unterklassen) etc. ab. Unabhängig von den Entscheidungen der **MO.Namen** (UUID oder (deutsche) Elementnamen) kann das OKSTRA-Modell auch nicht-deutschsprachigen Anwendern und Entwicklern sinnvoll zur Verfügung gestellt werden. Da die effiziente Verwendung z.B. in SPARQL-Abfragen jedoch auf den RDF-Knoten (und damit seiner deutschen Bezeichner) basiert, sollten auch langfristige Komplettübersetzungen (incl. Klassennamen und Beziehungsnamen etc.) angestrebt werden. Das parallele Vorhalten und bidirektionale Abbilden von verschiedenen Sprachen kann mit geeigneten Beziehungen wie **owl:sameAs** unterstützt werden.
- **Alignment mit anderen Vokabularen und Modellen.** Hinsichtlich des Mappings des OKSTRA Modells auf andere Vokabulare und Modelle spielen verschiedene Modellierungsoptionen unterschiedlich stark gewichtete Rollen. Für ein Verknüpfung von Klassen in okstraOWL beispielsweise mit dem CB-NL Core Vokabular⁵⁷ oder der Objekttypenbibliothek OTL⁵⁸ von Rijkswaterstaat (RWS) ist insbesondere die unterschiedliche Granularität und die zum Teil erheblichen Divergenzen in der Semantik ähnlich bezeichneter Objekttypen zu beachten.

⁵⁶ Beispielsweise dem konfigurierbaren Navigationsinterface des Werkzeuges TopBraid Composer

⁵⁷ <http://cbtnl.org/xwiki/bin/view/2.+Accessing+the+CB-NL/SPARQL-endpoint>

⁵⁸ <https://otl.rws.nl/publicatieomgeving/#/otl/rws-otl/library/otl-lib/version/2.0/object/ct0050/info>

2.4 AP 4: Übersetzung des OKSTRA in OWL

Auf Basis der im vorangegangenen Abschnitt dargelegten Abwägungen und Entscheidungen in Hinblick auf die Abbildung von OKSTRA auf OWL wurde in diesem Arbeitspaket eine entsprechende Konvertierung zur Generierung des okstraOWL-Schemas vorgenommen. Der Fokus lag dabei auf dem Schema *S_Entwurf*. Um dieses als OWL-Schema zu repräsentieren, mussten auch die Definitionen der Basisschema *okstra-basis.xsd*, *okstra-typen.xsd* sowie *S_Allgemeine_Geometrieobjekte* und *S_Allgemeine_Objekte* konvertiert werden.

Das entstandene Schema wurde ausführlich dokumentiert und wird auf dem Server der BAST als dereferenzierbarer SPARQL Endpoint bereitgestellt, was eine unmittelbare Nutzung des Schemas in SPARQL-Anfragen ermöglicht.

Daneben wurde ein Konverter für Instanzdatensätze entwickelt, der es erlaubt, OKSTRA-Instanzdaten der o.g. Schemata in entsprechende RDF-Datensätze zu überführen.

2.5 AP 5: Vergleich und Harmonisierung der Definition der Objekte und ihrer Attribute mit Anwendung von Linked Data

In diesem Arbeitspaket wurden die verschiedenen Modellierungsmöglichkeiten von semantischen Verknüpfungen mehrerer Modelle in Hinsicht auf ihre Einsetzbarkeit einer okstraOWL-<->CB-NL/OTL Kopplung untersucht. Ausgehend von der entwickelten OWL-Repräsentation des OKSTRA wurden Wege untersucht, um beide Modelle miteinander zu verknüpfen. Dazu wurde unter Zuhilfenahme von Linked Data Techniken (z.B. Silk Workbench⁵⁹) zunächst ein semi-automatisches Alignment der Modelle erzeugt, indem Äquivalenz-Beziehungen (*owl:sameAs*, *rdfs:seeAlso*, *skos:closeMatch* etc.) zwischen einzelnen Konzeptdefinitionen und deren Attributen definiert wurden. In einem zweiten Schritt wurden diese Beziehungen durch geeignete Werkzeuge manuell überprüft und bestätigt.

2.5.1 Mappings

Um unterschiedliche Vokabularien und Datenmodelle wie den OKSTRA und die Niederländische Object Type Library (OTL) bzw. andere nationale Standards, die unter der CB-NL erfasst sind, in einem gemeinsamen Datenraum zu überführen und zu verwenden, können für die derzeitigen Repräsentationen (OKSTRA: XML/GML; OTL: RDF und GML) zwei wesentliche Verfahrensansätze unterschieden werden:

- a) Die Konvertierung eines Instanzdatenformates in das jeweils andere (OKSTRA→OTL bzw. OTL→OKSTRA)
- b) Die Bereitstellung eines Werkzeuges, das jeweils beide Datenformate verarbeiten und so z.B. auch kombiniert abfragen kann.

Im Falle der Repräsentation des OKSTRA mithilfe von RDF-OWL (siehe APs 1-4) fällt für beide Herangehensweisen die Konvertierung in die jeweils andere Syntax weg, da sie sich beide in Gestalt von RDF Graphen bereites mit den gleichen Werkzeugen verarbeiten lassen. Diese Überwindung der **syntaktischen** Interoperabilitätshürden löst jedoch nicht das Problem ihrer **semantischen** Unterschiede: Wie im AP2 ausführlich erläutert und analysiert, sind beide Datenmodelle von sehr unterschiedlichem Umfang, Granularität und Topologie. Sowohl für die Konvertierung als auch für die gemeinsame Verarbeitung von Datensätzen im jeweiligen Format (und evtl. zusätzlicher zukünftiger anderer Formate z.B. in anderen Nachbarländern) müssen die Datenmodelle aneinander ausgerichtet (engl. *to align*, **the alignment**) werden. Hierzu können verschiedene Verfahren, teils automatisiert, angewendet werden.

In einem ersten Schritt, dem sog. **matching**, werden dazu Ähnlichkeiten zwischen verschiedenen Ressourcen der Modelle gesucht. Ähnlichkeiten zwischen verschiedenen Ressourcen können z.B. logischer Natur

⁵⁹ <http://silkframework.org>

sein: Zwei Konzepte C_1 und C_2 , die beide Unterkonzepte des Konzeptes P_1 sind, sind „miteinander verwandt“ oder „Geschwisterkonzepte“. Wenn sie sich in keiner Facette (Rollenbeziehung, Eigenschaft) unterscheiden oder nicht explizit als disjunkt (`owl:disjointWith`) ausgezeichnet sind, können sie sogar „ein und dasselbe“ sein, was sich in der OWL mit der Beziehung `owl:sameAs` ausdrücken lässt. Für Inferenzmaschinen bedeutet diese „identisch-mit“ Beziehung, dass alle anderen Facetten des Konzeptes C_1 auch für das Konzept C_2 gelten. Abgeschwächte Formen dieser Ähnlichkeitsbeziehungen lassen sich in anderen Vokabularien beispielsweise durch `rdfs:seeAlso` oder `skos:closeMatch` ausdrücken, ohne dabei streng formallogische Konsequenzen zu haben. Abbildung 39 stellt eine Übersicht semantisch unterschiedlich starker Ähnlichkeitsbeziehungen nach [Halpin 2010] dar. In diesen abgeschwächten Formen kann man eigene Abbildungsregeln entweder z.B. durch einfache Abfragen („ermittle alle Abschnitte oder Äste oder Instanzen von Konzepten, die diesen ähnlich sind“) ad hoc ermitteln oder mittels Regelsprachen als zusätzliche Informationen für die Grafen materialisieren („für alle Instanzen des `rdf:type ot1:IN00034` [Hoofdrijbaan, Main Carriageway, Hauptverkehrsstrasse], gilt, dass sie auch ein `okstra:Abschnitt_oder_Ast` sind“). Andere Ähnlichkeitsbeziehungen können beispielsweise Teil-Ganzes (part-of) Beziehungen oder Verbindungen sein.

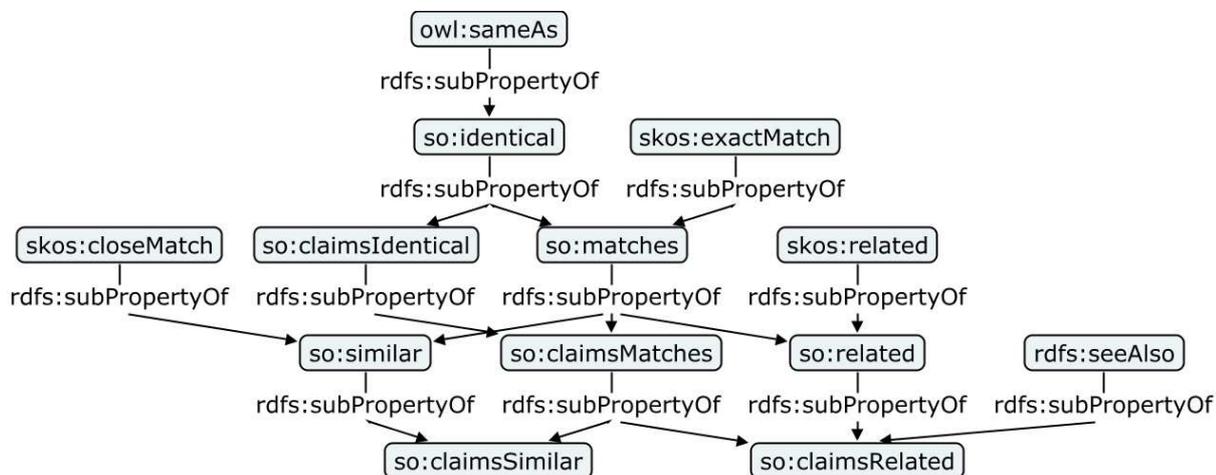


Abbildung 39 Unterschiedliche Ähnlichkeitsbeziehungen zwischen Konzepten [Halpin 2010]

2.5.2 Strukturähnlichkeiten

Da es bei umfangreichen Modellen wie dem OKSTRA und der OTL viele Tausend Konzepte und Rollenbeziehungen gibt, die sich zwar punktuell konzeptionell überschneiden, aber sehr unterschiedliche Detailniveaus und Bedeutungsebenen haben, sind starke Beziehungen wie `owl:sameAs` für Alignments eher ungeeignet, da sie viele unerwünschte Nebeneffekte mit sich bringen. Prinzipiell können automatisierte Alignment Werkzeuge anhand von Mustern in den Beziehungen unterschiedlicher Konzepte zueinander und den Rollenbeziehungen zu anderen Konzepten teilautomatisierte Verbindungen (prealignments) suchen. Die derzeit verfügbaren Werkzeuge wie COMA+⁶⁰, die solche Strukturähnlichkeitsmatchings unterstützen sind jedoch für derartig umfangreiche Datenmodelle wie OKSTRA und OTL nicht geeignet⁶¹.

Zur Illustration der divergierenden Struktur und Definitionstiefe der beiden Modelle sind in Abbildung 40 die OKSTRA-Konzepte `okstra:Strassenkante` und das OTL-Konzept `otl:RoadLink`⁶² mit ihren jeweiligen Subklassen dargestellt. Während Road Link zwölf Vererbungsstufen unter `owl:Thing` steht, sind es bei `okstra:Strassenkante` lediglich drei.

⁶⁰ <https://dbs.uni-leipzig.de/de/Research/coma.html>

⁶¹ Auf den verfügbaren Testmaschinen verfügbaren 32GB RAM reichten in verschiedenen durchgeführten Tests nicht aus.

⁶² Die korrekte Bezeichnung `otl:OB00025` wurde hier aus Gründen der Leserlichkeit gegen das englische Label ersetzt

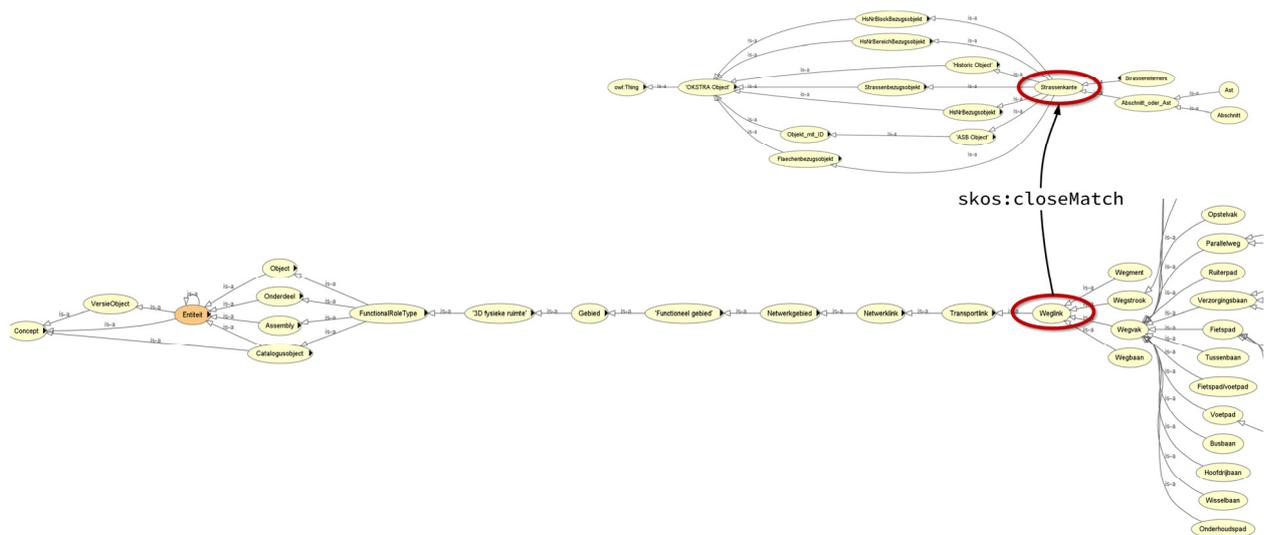


Abbildung 40 Verlinkung ähnlicher Konzepte durch eine skos:closeMatch Relation zwischen dem OKSTRA Konzept "Strassenkante" und dem OTL "Road Link" (Weglink)

Zusätzlich zu diesen topologischen Unterschieden in den Sub- und Superklasse-Beziehungen der jeweiligen Modelle, können zusätzlich Unterschiede auf Ebene der einzelnen Attribute ausgemacht werden. In Abbildung 41 ist schematisch dargestellt, wie Eigenschaften / Attribute zwischen den oben verwendeten `okstra:Strassenkante` und `otl:RoadLink` Konzepten aufeinander abbildbar sind. Hier wird deutlich, wie die Attribute `Object_ID`, `Liniengeometrie` und `Nummer_gehört_zu_Strasse`, die im OKSTRA auf verschiedene Klassen verteilt sind, im Konzept `otl:RoadLink` mittels Restriktionen bzw. domain-range-Einschränkungen verlinkt sind.

Abbildung 41 Mappings zwischen Eigenschaften des `otl:RoadLink` Konzeptes (Ausschnitt rechts) zu den äquivalenten Eigenschaften im OKSTRA Modell (links)

2.5.3 Sprachähnlichkeiten

Alternativ zu der Suche nach strukturellen Ähnlichkeiten können Ähnlichkeitssuchen für Bezeichner (labels) von Konzepten herangezogen werden. Im Zuge dieses Projektes wurden Tests durchgeführt, in denen die deutschsprachige `okstraOWL`-Version zunächst mit englischen Übersetzungen angereichert wurde. Die Basis dieser Arbeit bildete die Teilübersetzungen des OKSTRA ins Englische⁶³. Durch die Verwendung

⁶³ http://www.okstra.de/forschung/okstra_internationalization_EN.html

von Wortähnlichkeitsalgorithmen wie der Levenshtein-Distanz können so grobe Zuordnungen zwischen den verschiedenen Klassen vorgenommen werden, die sich dann mithilfe der oben beschriebenen Beziehungen beschreiben und weiternutzen lassen. Abbildung 42 zeigt die Erstellung eines Mapping Musters, Abbildung 43 illustriert Teilergebnisse dieses auf OKSTRA und OTL angewendeten Verfahrens. Aufgrund der vorhandenen Datengrundlage und insbesondere der Fokussierung der Übersetzungen auf das Teilschema Entwurf mit rund 450 übersetzten Konzepten und Eigenschaften, sind diese automatisierten Verknüpfungen einzelner Konzepte zwar ein hilfreicher Start, bedürfen aber für ein vollständiges, praxistaugliches Mapping sehr viel manueller Nacharbeit und der gezielten Erzeugung von Verknüpfungen durch Experten. Exemplarische Verknüpfungen wurden für einzelne Szenarien des AP 6 vorgenommen, um die mögliche Verwendbarkeit zu untersuchen.

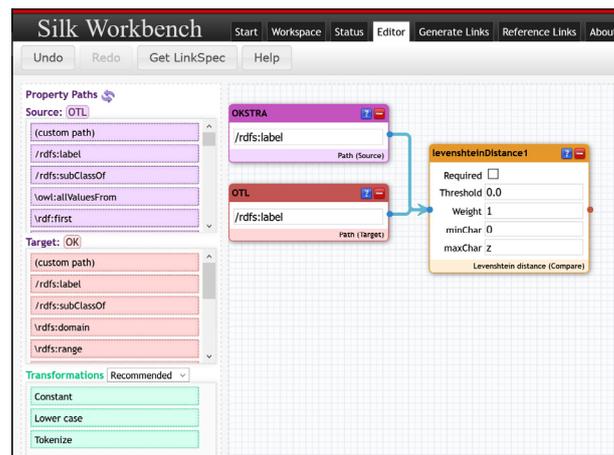


Abbildung 42 Konzept Matching zwischen okstraOWL und OTL mittels Sprachähnlichkeiten in der SILK Workbench

Source: OTL	Target: OK	Score	Correct
http://otl.rws.nl/mc-m0006#ct0067 Comparison: levenshteinDistance (levenshteinDistance1) 100.0% Input: rdfs:label (OKSTRA) Unit Input: rdfs:label (OTL) Einheit Unit	http://okstraowl.org/def/2017/okstraowl#Einheit	100.0%	<input checked="" type="checkbox"/>
http://otl.rws.nl/otl#OB01069	http://okstraowl.org/def/2017/okstraowl#Gewaesserart.06	100.0%	<input checked="" type="checkbox"/>
http://otl.rws.nl/otl-orgtype#otl-orgtype-person Comparison: levenshteinDistance (levenshteinDistance1) 100.0% Input: rdfs:label (OKSTRA) Person Input: rdfs:label (OTL) Person	http://okstraowl.org/def/2017/okstraowl#Person	100.0%	<input checked="" type="checkbox"/>
http://otl.rws.nl/otl#OB02860 Comparison: levenshteinDistance (levenshteinDistance1) 100.0% Input: rdfs:label (OKSTRA) Asphalt Asfalt Input: rdfs:label (OTL) Asphalt	http://okstraowl.org/def/2017/okstraowl#Oberflaeche_befestigte_Flaeche.01	100.0%	<input checked="" type="checkbox"/>
http://otl.rws.nl/otl#OB01007	http://okstraowl.org/def/2017/okstraowl#Hindernisart.1	100.0%	<input checked="" type="checkbox"/>
http://otl.rws.nl/otl#OB01034	http://okstraowl.org/def/2017/okstraowl#Habitat	100.0%	<input checked="" type="checkbox"/>

Abbildung 43 Teilergebnisse der Ähnlichkeitssuche von Konzepten

2.6 AP 6: Aufbereitung und Anreicherung von RWS-Datenbeständen mit OKSTRA-Inhalten

Für eine Untersuchung der möglichen Nutzung der Modelle in jeweils anderen Kontexten wurden die im AP 5 beschriebenen inhaltlichen Betrachtungen von Mappings anhand mehrerer von der BAST und RWS zur Verfügung gestellter Datensätze für ausgewählte Szenarien untersucht. Dabei handelt es sich jeweils um ein Netzwerkmodell der Hauptverkehrsstraßen von Nordrhein-Westfalen (grünfarbiger rechter Teil in Abbildung 44), sowie um das Autobahnnetz der Provinz Limburg (linker Teil in Abbildung 44), die an verschiedenen Stellen ineinander übergehen. Abbildung 45 zeigt eine Ausschnittsvergrößerung der grafischen Darstellung dieses gemeinsamen, grenzübergreifenden Datenraumes.

Hierin wird exemplarisch und grafisch der unterschiedliche Charakter und Informationsgehalt beider Datensätze deutlich: Die NRW-Daten enthalten vor allem topologische Netzdaten und bestehen zum überwiegenden Teil aus einzelnen Abschnitt-Elementen, die einzelne Kurvengeometrien als GML-Repräsentationen enthalten. Der niederländische Datensatz ist hingegen auf wenige Einzeltrassen beschränkt, weist jedoch einen höheren Detaillierungsgrad auf und enthält bspw. ausgiebige Daten zu Gelände, Beschilderungen, Beleuchtung, Kameras etc.

Im Gegensatz zu den OKSTRA-Daten, in denen geometrische Repräsentation und andere, z.B. alphanumerische Daten in einem Graphen zusammengefasst sind, setzen sich die RWS-OTL Daten aus einzelnen Abschnitten des COINS⁶⁴-Containers zusammen, der wie folgt aufgebaut ist:

Der zentrale Container besteht im Wesentlichen aus zwei Verzeichnissen (bim, doc), in denen sich unter bim/ eine zentrale RDF/OWL Datei (serialisiert im TURTLE Format) befindet, die die verschiedenen semantischen Instanzeinheiten definiert und dabei drei Hauptvokabulare referenziert (die selbstdokumentierend ebenfalls dem Container beiliegen): der RWS-OTL⁶⁵, dem COINS 2.0 Kernmodell sowie das Referenzrahmenwerk für den Straßenbau⁶⁶. Die geometrischen Repräsentationen sind im GML-XML Format einzeln nach Objekttypen im /doc/ Verzeichnis verlinkt. Einzelne Instanzobjekte im Datensatz werden mit Hilfe von Verweisen auf die jeweiligen GML IDs mit den semantischen Informationen verlinkt. Auf diese Weise können bidirektionale Abfragen (semantische Informationen zu gegebenen GML-Geometrien und geometrische Repräsentationen von Instanzdaten im RDF-Graphen) mithilfe von SPARQL erfolgen. Zu Testzwecken entstand dafür ein Softwarewerkzeug, das für die grafisch ausgewählten Objekte in einem GIS-Viewer semantische Informationen von einem SPARQL-Endpoint abrufen bzw. umgekehrt, Abfragen nach bestimmten Kriterien und Eigenschaften geometrisch visualisieren kann. Das folgende Listing gibt für das gewählte Objekt die im RWS-OTL Datensatz hinterlegten Informationen aus:⁶⁷

```
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix coins-base: <http://www.coinsweb.nl/cbim-2.0.rdf#>

SELECT DISTINCT ?prop ?val ?representation ?desc ?something ?parent
WHERE {

    ?representation coins-base:documentFragment ?prop .
    ?object coins-base:objectValue ?representation .
    ?parent coins-base:hasProperties ?object .
```

⁶⁴ COINS ist ein niederländisches Standardformat für die Übergabe von Planungsdaten im Infrastrukturbau an den Bauherrn und konzeptionelle sowie strukturell ähnlich zum Information Container Data Drop (ICDD), der sich momentan als ISO 21597 im internationalen Standardisierungsprozess befindet

⁶⁵ im vorliegenden in der Version 1.2

⁶⁶ rws-coins-20-referentiekader-2.0.ttl

⁶⁷ die Variable %s wird zur Laufzeit durch die im Viewer gewählte GML ID ersetzt

```

?parent coins-base:description ?desc .

?prop coins-base:datatypeValue ?gml_id .
FILTER (regex (?gml_id, "%s", "i")) .

}

```

2.6.1 Fallbeispiel A4-A76/E314

Um die heterogenen NRW und RWS-OTL Datensätze gemeinsam zu nutzen, können beide Datensätze durch die jeweils anderen Vokabulare zusätzlich ausgezeichnet werden. Für einen Proof-of-Concept wurde dazu der RWS-OTL Datensatz der Provinz Limburg mit okstraOWL Konzepten angereichert. Da in den zur Verfügung stehenden Daten auf der niederländischen Seite netzwerktopologische Verbindungen nur indirekt zur Verfügung standen, wurde eine Netzwerktopologie gemäß dem OKSTRA exemplarisch für die A76 angelegt. Die A76 verläuft auf der niederländischen Seite von der deutschen Grenze am Übergang Vetschau, wo sie aus der deutschen A4 übergeht durch Limburg über Heerlen bis an die belgische Grenze über die Maas bei Nieuwdorp, wo sie als E314 weiterläuft.

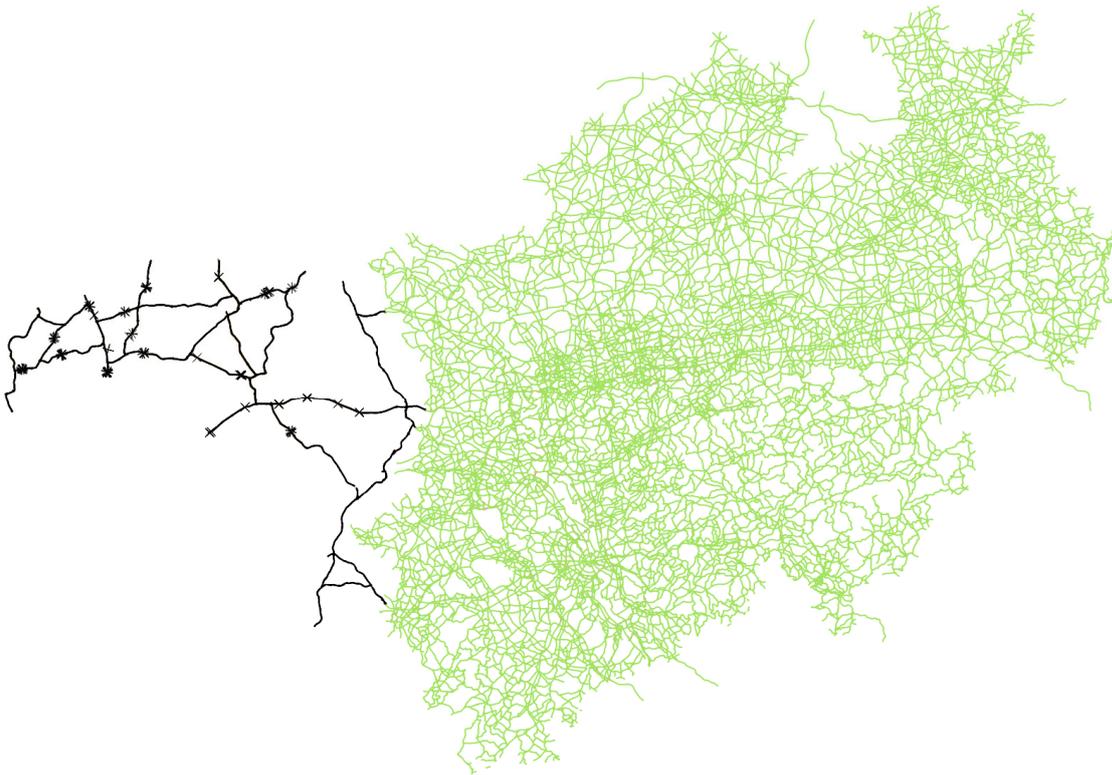


Abbildung 44: Graphische Übersicht eines gemeinsamem Datenraumes aus niederländischen (schwarz) und deutschen (grün) Straßendaten aus verschiedenen Quellen in verschiedenen Datenformaten

Für die technische Umsetzung wurden die 18 einzelnen Abschnitte der A76/Rijksweg 76, die im RWS-OTL Vokabular mit dem Konzept `ot1:IN00025` („Roadlink“, „Wegvak“) bzw. `ot1:IN00034` („Hoofdrijbaan“, „Main carriageway“) ausgezeichnet werden, als `okstra:Abschnitt` klassifiziert. Da im niederländischen Datensatz ein Pendant zum Konzept `okstra:Nullpunkt` nicht vorhanden ist, wurden Kreuzungspunkte der A76 auf der jeweils „linken“ Seite (von Belgien Richtung Deutschland in südöstliche Richtung laufend) gewählt. Diese sind im RWS-OTL Datensatz als `ot1:OB01294` („Ongelijkvloerse kruising van netwerklinks“, „Multi-level crossing of network links“) ausgezeichnet und geben Kreuzungen auf verschiedenen Niveaus

(Unter- oder Überführungen) an. Um diese als Start- und Endpunkt der jeweiligen Abschnitte analog zum NRW-Datensatz behandeln zu können, wurden sie zusätzlich als `okstra:Nullpunkt` klassifiziert. Am Grenzübergang (siehe Abbildung 45) wurde ein weiterer, „virtueller“ Abschnitt angelegt, der den letzten niederländischen Nullpunkt mit dem ersten deutschen Nullpunkt verknüpft:

```
<http://dataroom.otl.rws.nl/otl171#399030011_wvk> a okstra:Abschnitt ;
  rdfs:label "UEBERGANG_LANGVELD" ;
  okstra:beginnt_bei_NP__Abschnitt_oder_Ast
    <http://dataroom.otl.rws.nl/otl171#399030003_wvk-kruist-399030002_wvk> ;
  okstra:endet_bei_NP__Abschnitt_oder_Ast
    <http://bast.de/nrw/ins_nrw#Oklabi.15928.090001hlp7k>;
  rdfs:comment "R76_026.308_L_n" .
```

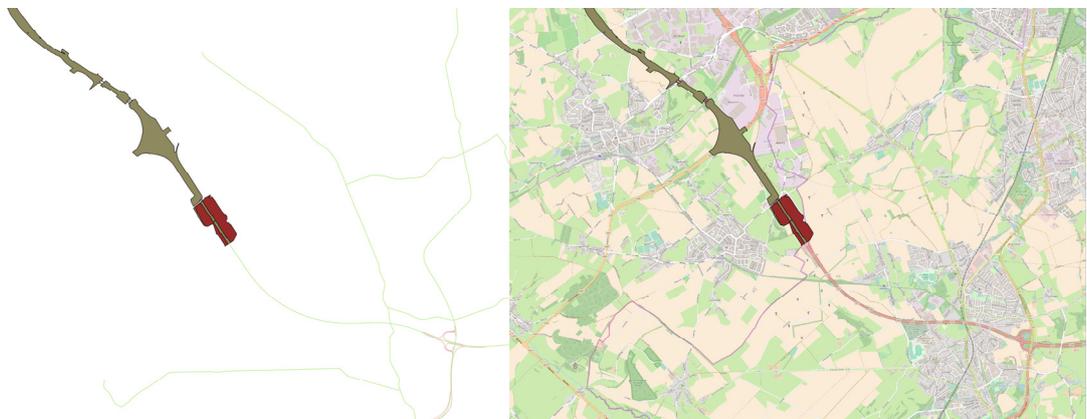


Abbildung 45: Überlagerte Datensätze RWS-OTL "Dataroom" OKSTRA NRW sowie zusätzliche OpenStreetMap Überlagerung am Grenzübergang Vetschau an dem die Deutsche A4 in die niederländische A76 übergeht.

Um unter „realistischen“ Bedingungen getestet zu werden, wurden diese zusätzlichen Informationen mithilfe von SPARQL CONSTRUCT queries aus den Originaldatensätzen erzeugt und in separaten Graphen in demselben Triple Store abgelegt, in dem auch alle anderen Datensätze vorgehalten werden. SPARQL Suchanfragen können dann, auch über Server-Grenzen hinaus, über alle Graphen hinweg gestartet werden. In Abbildung 46 ist die Übersicht der für die Tests verwendeten quelloffenen Triple-Store Plattform Apache Jena Fuseki zu sehen, die ebenfalls Statistiken über die Umfänge der verwendeten Datensätze darstellt. Der Graph „okstradata“ ist mit 14 Mio. Tripeln noch vor dem RWS „dataroom“ Datensatz mit rund 3,9 Mio. Tripeln der größte, lässt sich aber auf aktueller Mittelklasse-Hardware⁶⁸ für die meisten Anwendungsszenarien performant verwenden.

⁶⁸ Intel i7 2,9 GHz, 32GB RAM

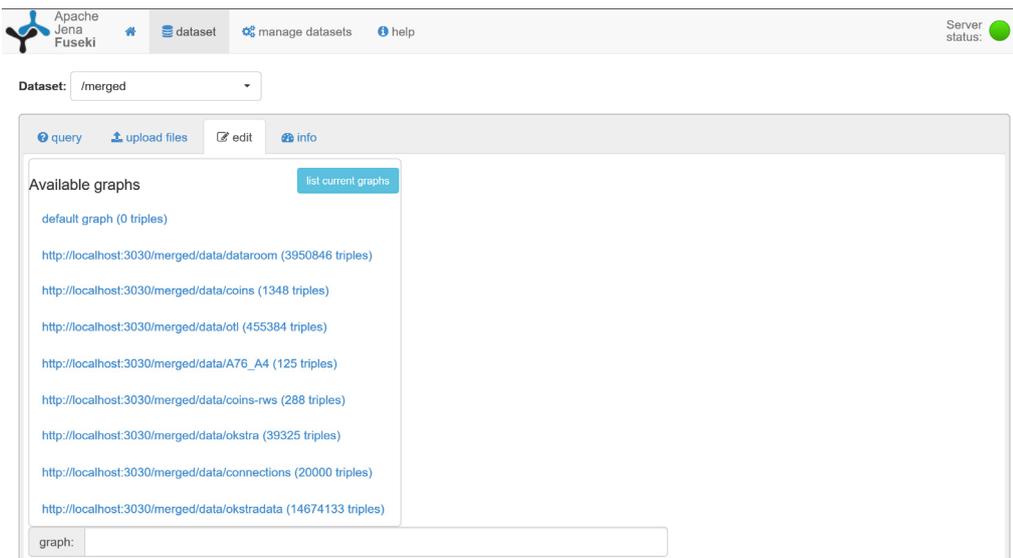


Abbildung 46: Übersicht der verschiedenen Grafen die zu Proof-of-Konzept Zwecken in einen gemeinsamen Datenraum gelegt wurden.

Um in Testszenarien Eigenschaften der jeweiligen Abschnitte, wie bspw. Zustandsdaten, Abmessungen oder Kamerastandorte entlang einer bestimmten Strecke wählen zu können, sind die indirekten Verbindungen zwischen Knoten und Kanten im Straßennetz, wie sie in OKSTRA als `okstra:Abschnitt. beginnt_bei_NP__Abschnitt_oder_Ast → okstra:Nullpunkt ← okstra:endet_bei_NP__Abschnitt_oder_Ast` modelliert sind, für effiziente und vor allem rekursive SPARQL-Anfragen eher ungünstig. Um einzelne Abschnitte direkter miteinander zu verbinden, ohne den Umweg der Nullpunkte wählen zu müssen, sind durch die folgende SPARQL CONSTRUCT Anfrage weitere, direkte und explizite Links zwischen den einzelnen Abschnitten angelegt („im Graphen materialisiert“) worden:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX otl-base: <http://otl.rws.nl/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX coins-base: <http://www.coinsweb.nl/cbim-2.0.rdf#>
PREFIX okstra: <http://okstraowl.org/def/2017/okstraowl#>

CONSTRUCT
  ?s2 okstra-plus:beginnt_bei ?s .
  ?s okstra-plus:endet_in ?s2
}

FROM Named <http://localhost:3030/merged/data/dataroom>
FROM Named <http://localhost:3030/merged/data/okstradata>
FROM Named <http://localhost:3030/merged/data/okstra>
  FROM Named <http://localhost:3030/merged/data/otl>
  FROM Named <http://localhost:3030/merged/data/A76_A4>

WHERE {
  GRAPH ?c1 {
    ?s okstra:beginnt_bei_NP__Abschnitt_oder_Ast ?np .
  }
}

```

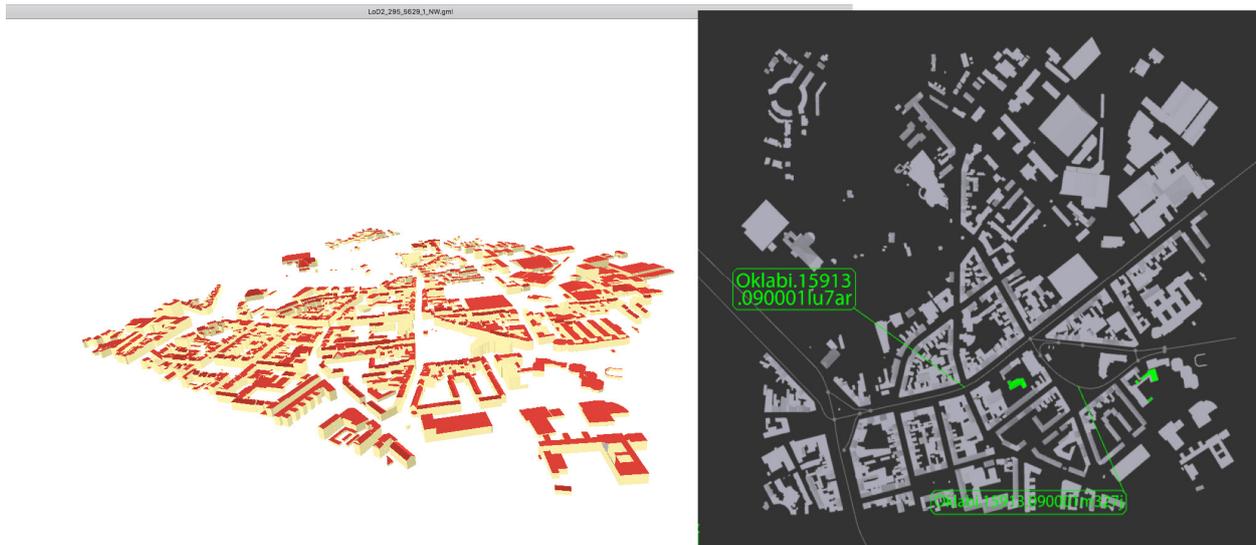



Abbildung 48: Gemeinsame Darstellung von 3D CityGML Daten mit OKSTRA Geometrien und Informationen

Die anfänglichen gemeinsamen Informationen beider Ausgangsdatensätze sind in dieser Ausgangssituation zunächst auf die räumlich-geometrischen Ebenen beschränkt, die Abfragen selbst mit Hilfe von GeoSPARQL nur wenig effizient zulassen. In einem weiteren Arbeitsschritt wurden daher automatisiert Links zwischen den verschiedenartigen Objekten beider Graphen generiert, die sich die impliziten räumlich-geografischen Beziehungen zu Nutze machen. Dazu wurden zunächst die Gebäude-Standflächen der dreidimensionalen Repräsentation aller Gebäude aus dem Gesamtkörper gelöst und anschließend ihre geometrischen Mittelpunkte errechnet. Die GeoSPARQL-Funktion `geof:distance` kann dafür herangezogen werden und erlaubt so, den kürzesten Abstand zwischen CityGML-Gebäude und dem jeweiligen OKSTRA-Abschnitt zu ermitteln:

```
SELECT ?a (MIN(distance) AS ?mindis] ?b
WHERE {
  ?a rdf:type bldg:building
  ?a geo:hasGeometry ?ageo .
  ?ageo geo:asWKT ?awkt .
  ?a bldg:function ?afunction .
  ?b geo:asWKT ?bgeo .
  ?b rdf:type okstra:Abschnitt
  ?bgeo geo:asWKT ?bwkt .

  BIND (geof:distance (?awkt, ?bwkt, uom:meter) AS ?distance) .

  FILTER (?ageo != ?bgeo)
}
```

Aus den so entstehenden Listen der räumlich nächsten `okstra:Abschnitte` können dann explizite Links zwischen Gebäuden und Straßen angelegt werden, die nun auch ohne entsprechende GeoSPARQL Implementierungen nutzbar sind.

Zusammen mit den vielfältigen Informationen, die im CityGML-Modell enthalten sind, wie etwa der Nutzungsart oder der Höhe der jeweiligen Gebäude, kann der so entstandene gemeinsame OKSTRA/CityGML Informationsraum für Fragestellungen wie „Wo befinden sich Kindergärten und Schulen im Umkreis von 100 m in der Nähe zu einer Bundesstrasse“ (Abbildung 49 a)) oder „Welche Gebäude mit einer Höhe von über 20 m sind näher als 50 m an einer Schnellstraße?“ (Abbildung 49 b)) genutzt werden.

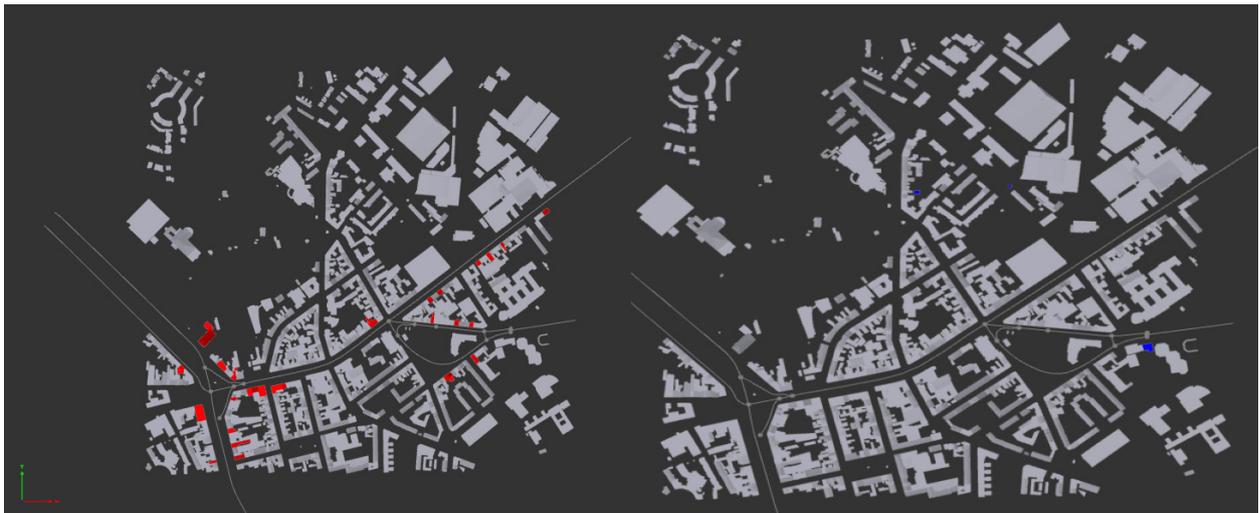


Abbildung 49: Visualisierungen von (a) Schulen und Kindergärten in 100 m Radius und b) Gebäuden mit einer Höhe > 20 m

Für die Präsentation der mit SPARQL ermittelten Ergebnisse wurde mit Hilfe eines in der kommerziellen Software FME Desktop erstellten Skriptes die jeweiligen CityGML-Objekte rot bzw. blau eingefärbt und als 3D-Modell in ein PDF eingebettet (Abbildung 50).

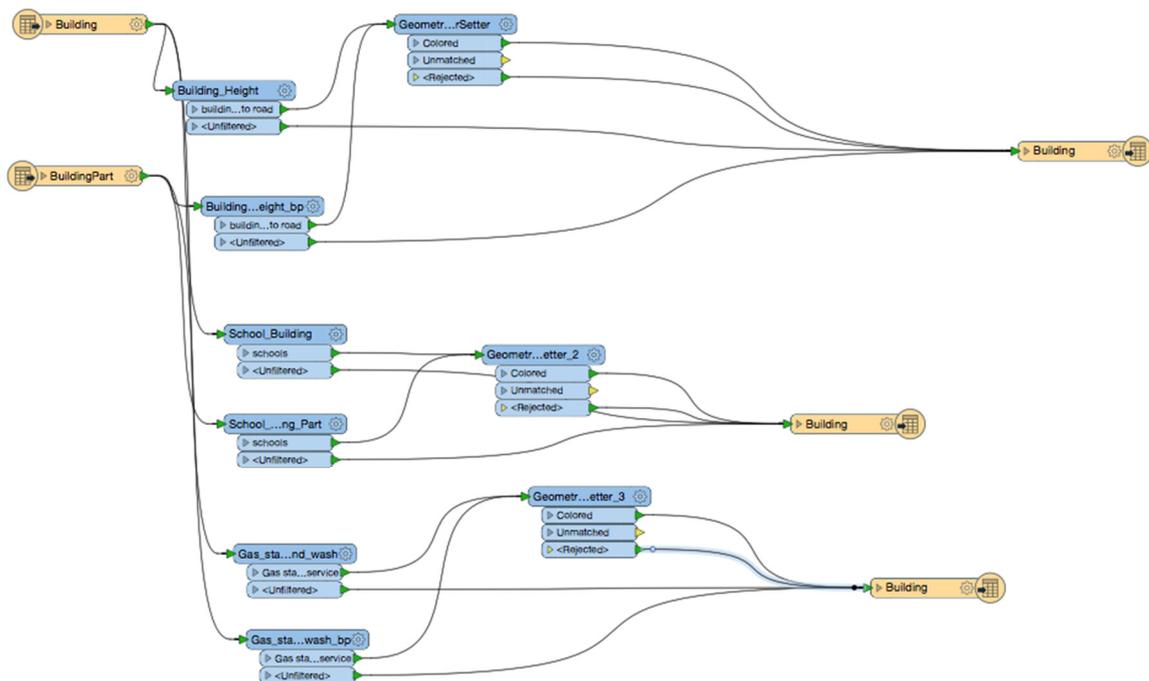


Abbildung 50: FME Skript zum Einfärben der SPARQL Anfrageergebnisse

Die hier beschriebenen Szenarien zur Verlinkung und Anreicherung von Information durch okstraOWL illustrieren die vielversprechenden Anwendungen, die mithilfe breiter Unterstützung von standardkonformen Werkzeugen aus dem Bereich Linked Data flexibel und verteilt realisiert werden können.

Da die Qualität und Informationstiefe der zur Verfügung stehenden Testdatensätze höchst unterschiedlich und teilweise auf kleinere geografische Räume beschränkt war, lassen sich über die allgemeine Performanz und Skalierbarkeit der gezeigten Ansätze nur grobe Einschätzungen treffen. Die Datengrößen im Bereich von zweistelligen Millionen Tripeln sind auf herkömmlicher Hardware für einfachere Anfragen gut

handhabbar, d.h. es ergeben sich Antwortzeiten von wenigen Sekunden (DBPedia besteht derzeit aus ca. 3 Billionen Tripeln), können jedoch für kompliziertere Fragestellungen, wie etwas rekursive Suchen in Netzwerktopologien mit SPARQL Property Paths schnell „explodieren“. Der deutliche Vorteil der Herangehensweise mit Linked Data gegenüber konventionellen Datenbanklösungen liegt in der einfachen Handhabbarkeit, leichten Erweiterbarkeit, Verteilbarkeit und Flexibilität der Erstellung der Abfragen. In den Tests konnte gezeigt werden, wie auf einfache Weise Strukturen des semantisch reichen OKSTRA-Modells herangezogen werden können, um andere Daten semantisch anzureichern und über Domänengrenzen hinweg einheitliche anwendbar zu machen.

2.7 AP 7: Erarbeitung weiterführender Modellverwendungen

In diesem Arbeitspaket wurden Lösungsansätze für eine modellgestützte iterative Ausarbeitung eines Straßenbauprojektes mit Hilfe der vorher entwickelten Modellierungsmethoden erarbeitet.

Die Verfügbarkeit des OKSTRA-Modells im Format okstraOWL erlaubt die Anwendungen einer ganzen Reihe von Funktionalitäten des Semantic Web. Dabei steht zunächst die Möglichkeit der Abfrage von Informationen mit der Sprache SPARQL im Vordergrund. Damit ist zunächst die Möglichkeit der umfassenden Analyse von OKSTRA-Daten mithilfe standardisierter Werkzeuge möglich. Dies umfasst alle Bereiche, die von OKSTRA abgedeckt werden, vom Entwurf bis zur Zustandserfassung und –bewertung. Die Sprache erlaubt es darüber hinaus insbesondere, Datenquellen aus anderen Domänen (und anderen Datenmodellen) einzubeziehen.

Neben der in AP 6 untersuchten integrativen Anfragen von OKSTRA und CB-NL gibt es eine Fülle von weiteren Anwendungsmöglichkeiten in diesem Kontext. Dazu gehört zum Beispiel die integrative datenmodellübergreifende Formulierung und Beantwortung von Anfragen, bei denen in CityGML vorliegende 3D-Stadtmodelle zusammen mit OKSTRA-Datensätzen ausgewertet werden, um beispielsweise Fragen der Lärmimmission zu beantworten. Weitere interessante Anwendungsfelder liegen in der Zusammenführung von Unfall- mit Straßenzustands- und Wetterdaten, um bspw. mögliche Korrelationen zwischen der fehlenden Rauigkeit des Straßenbelags und dem Auftreten von Unfällen zu identifizieren.

2.7.1 Verknüpfung von OKSTRA-Bestandsdaten und OKSTRA-Zustandsdaten

Mit Vorliegen von okstraOWL ist es möglich, Objekte des Schemas *S_Strassennetz* mit Objekten des Schemas *S_Strassenzustandsdaten* zusammenzuführen. Bisläng liegen diese Objekte ohne Verknüpfungsmöglichkeit in getrennten Schemata vor.

Der untersuchte Beispieldatensatz aus NRW (OKSTRA-Version 1.012) beinhaltet zum großen Teil Instanzen vom Typ *Ast* bzw. *Abschnitt*, welche jeweils Subklassen der Klasse *Abschnitt_oder_Ast* sind. Mit jedem *Ast* wird eine Geometrie in Form eines linienförmigen Objektes (*GM_Curve*) verknüpft. Im Datensatz liegen zudem Informationen in Form von Objekten des Typs *Querschnittstreifen* vor, die mit den Objekten vom Typ *Teilabschnitt* oder *Ast* über das Attribut *hat_Strecke* verknüpft sind.

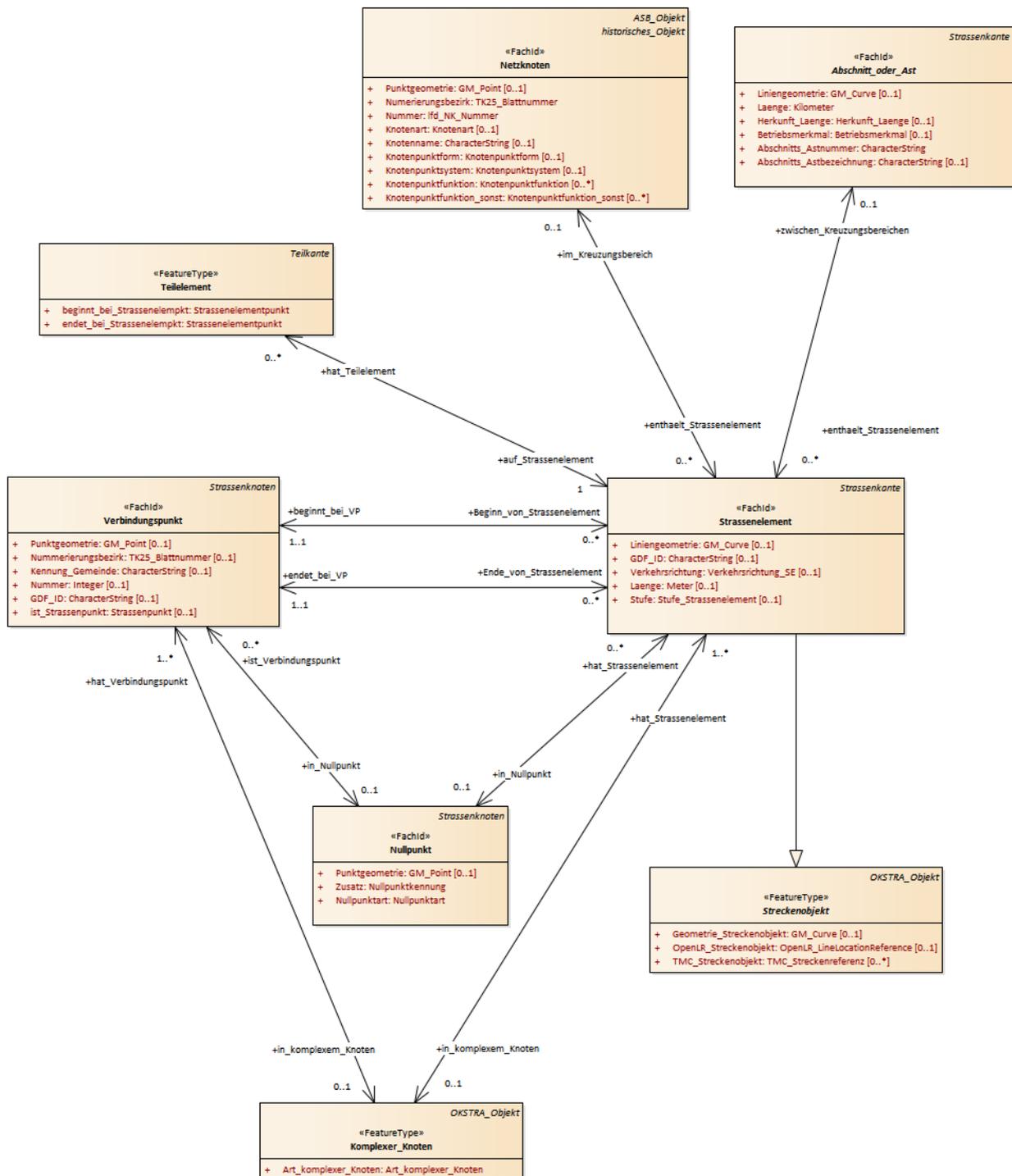


Abbildung 51: Objekte des Schemas S_Strassennetz: Die Klasse Abschnitt_oder_Ast trägt die Liniengeometrie der Bestandsdaten

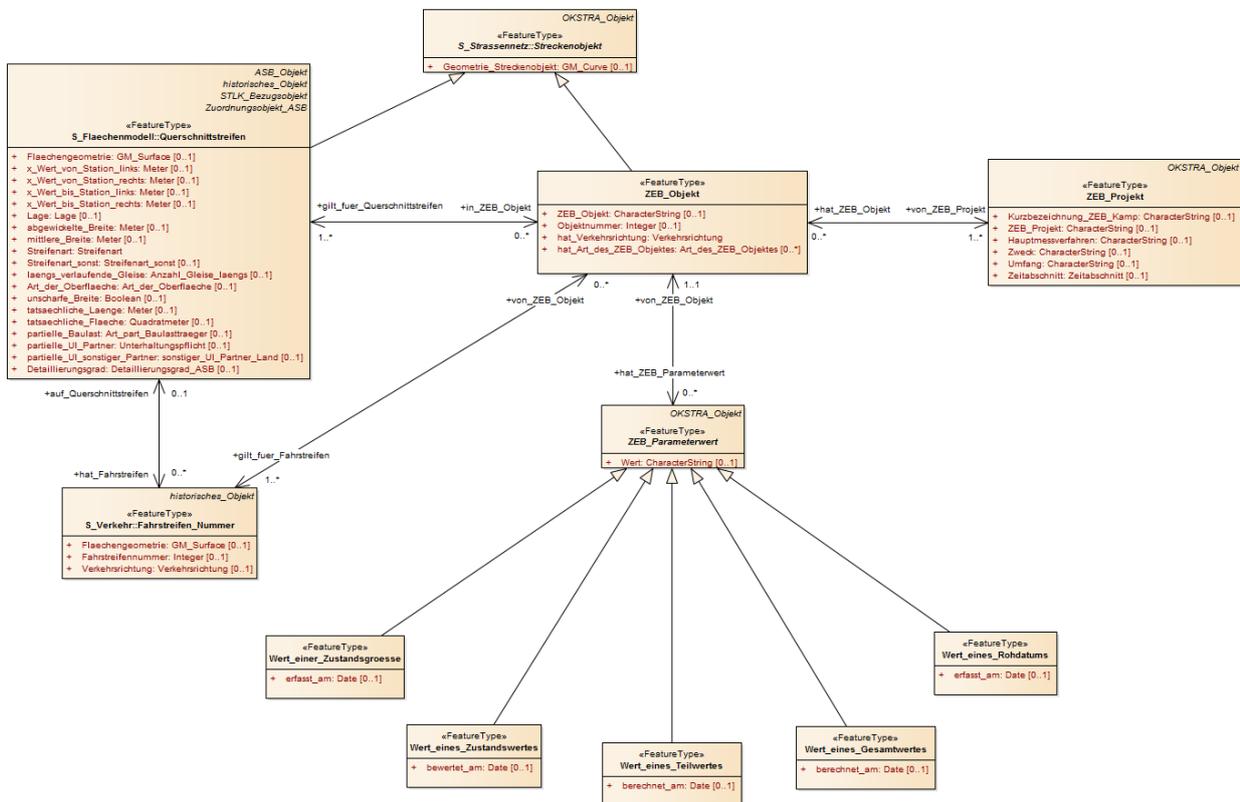


Abbildung 52: Ausschnitt des Schema S_Zustandsdaten

Wichtigstes Element der Straßenzustandsdaten ist die Klasse *ZEB_Objekt*. Instanzen dieser Klasse beschreiben den Gegenstand einer Zustandserfassung bzw. –bewertung. Die räumliche Lage eines *ZEB_Objektes* korrespondiert mit der Lage eines Erfassungsabschnittes. Die Lage wird beschrieben durch die Superklasse *Streckenobjekt*, die linienförmige Kurven vom Typ *GM_Curve* repräsentieren kann. Ein Erfassungsabschnitt hat eine achsenbezogene Ausdehnung (*Streckenobjekt*) sowie eine querschnittsbezogene Ausdehnung. Im Straßenquerschnitt ist das *ZEB_Objekt* durch einen oder mehrere Fahrstreifen bestimmt. Dies wird im Datenmodell dadurch modelliert, dass *ZEB_Objekt* ein oder mehrere Objekte vom Typ *Querschnittstreifen* referenziert.

Die verwendeten GML-Geometrien bieten eine hervorragende Grundlage für die Verknüpfung zusammengehöriger Bestands- und Zustandsdaten. Mithilfe der Funktionalitäten der Anfragesprache GeoSPARQL, die auf Basis von GML-Geometrie räumliche Analyseverfahren anbietet, lassen sich durch topologische Prädikate als Kriterium für die Verknüpfung bzw. gemeinsame Auswertung heranziehen. So kann beispielsweise der Linienzug eines *ZEB_Objekt* auf räumliche Nähe mit dem Linienzug eines Bestandsobjektes geprüft werden.

Eine Möglichkeit liegt dabei in der Bildung eines räumlichen Puffers um das ZEB-Linienobjekt und Überlagerung mit den *Abschnitt*-Objekten des Bestandsdatensatzes. Insbesondere kann hier die Lage der Polygonpunkte (*GM_Point*) auf Überlappung mit dem gebildeten Puffer geprüft werden, um auf diese Weise eine Verknüpfung herzustellen.

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
```

```
SELECT ?what
WHERE {
  ?what geo:hasGeometry ?geometry .
```

```

FILTER(geof:ehCovers(?geometry,
  "POLYGON((
-77.089005 38.913574,
-77.029953 38.913574,
-77.029953 38.886321,
-77.089005 38.886321,
-77.089005 38.913574
))"^^geo:wktLiteral))
}
    
```

Das gleiche gilt analog für die Objekte des Schemas S_Bauliche_Strasseneigenschaft. Auch hier sind bspw. die Objekte vom Typ *Aufbauschicht* mit einer entsprechenden GML-Geometrie versehen, die für die oben beschriebene räumliche Verknüpfungsmethode herangezogen werden können. Bohrkern verfügen über eine GML-Punktgeometrie und können entsprechend ebenfalls mithilfe räumlicher Abfragen verknüpft werden.

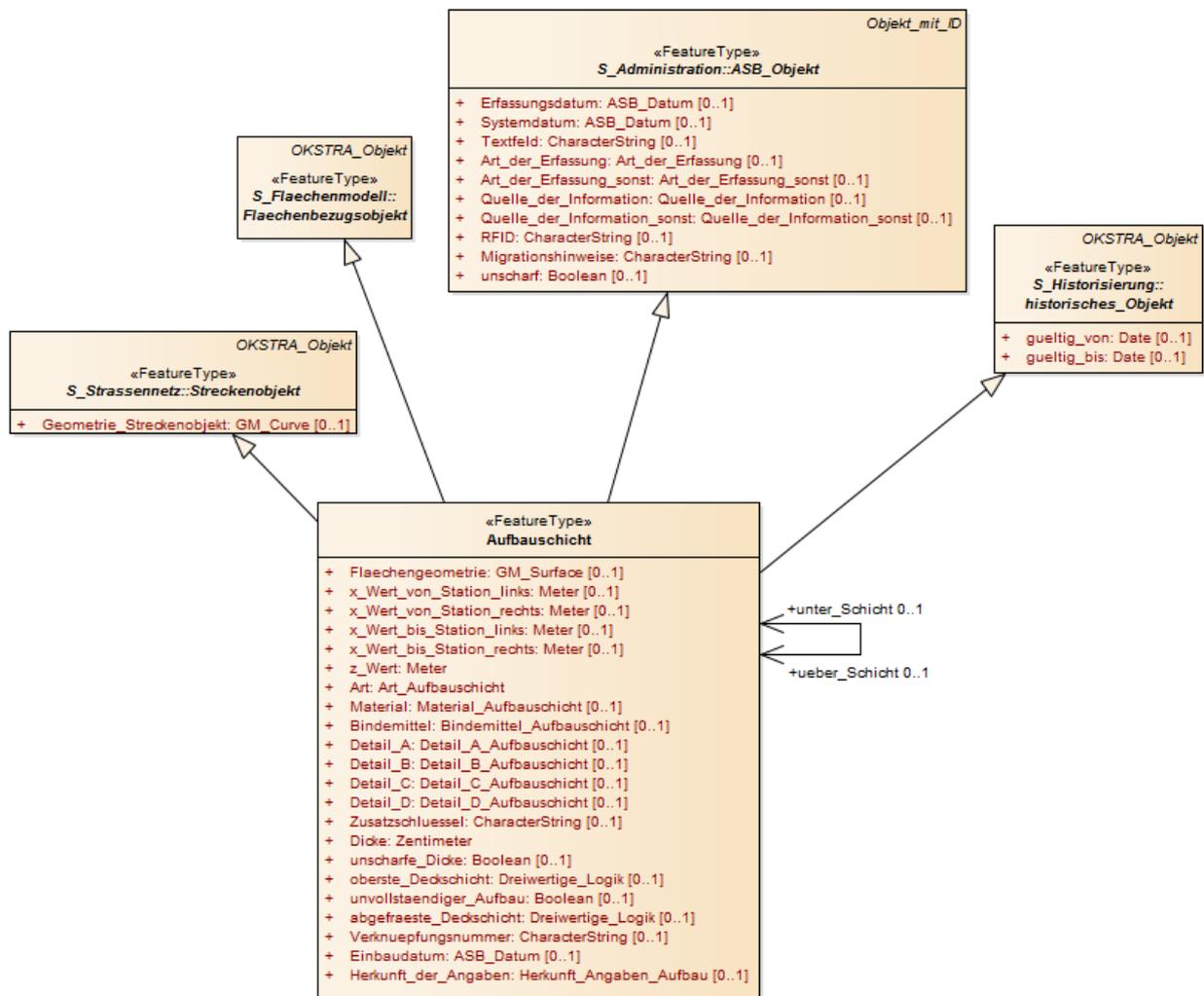


Abbildung 53: Ausschnitt aus dem Schema S_Bauliche_Strasseneigenschaften

Durch einen ähnlichen Mechanismus lassen sich auch Entwurfs- und Zustandsdaten miteinander verknüpfen.

2.8 AP 8: Prototypische Implementierung einer gemeinsamen Datenmodellierung und Repräsentation von OKSTRA-Objekten

Die TUM Open Infra Platform wird am Lehrstuhl für Computergestützte Modellierung und Simulation der Technischen Universität München entwickelt. Die Entwicklung der Software begann bereits im September 2013 und wird im Rahmen von Forschungs- und Drittmittelprojekten kontinuierlich weitergeführt. Die Software wird kostenlos zur Verfügung gestellt und kann von der Lehrstuhlwebseite heruntergeladen werden⁷². Der Quellcode der Software ist ebenfalls unter einer Open Source Lizenz verfügbar (GPL v3) und wird auf Bitbucket⁷³ und GitHub⁷⁴ bereitgestellt. Im Vordergrund der Software steht die Förderung von Open-BIM-Ansätzen für die Infrastrukturplanung durch die prototypische Implementierung verschiedener offener Bauwerksdatenmodelle. Dabei werden Standards, wie OKSTRA, Datenart 40, LandXML, InfraGML oder etwa IFC 4.1 unterstützt. Daneben unterstützt die Software auch typische Elemente des Hochbaus (Türen, Fenster, Wände, etc.) und Teile des IFC2.3 Standards. Durch die Offenheit der Software kann diese auch als Referenzimplementierung herangezogen werden und so die Integration von Standards in andere Softwareprodukte beschleunigen.

Die TUM Open Infra Platform (kurz OIP) wurde im Rahmen dieses Projekts um Exportmöglichkeiten für das Schreiben von okstraOWL-Instanz-Dateien erweitert. Dabei wurde die OKSTRA-Klassenbibliothek (OKLABI) genutzt, die kostenfrei von der OKSTRA-Webseite⁷⁵ bezogen werden kann. Diese wird genutzt, um OKSTRA-Instanzdaten einzulesen. Für das Schreiben von OWL-konformen Dateien wurde die Bibliothek Raptor RDF Syntax Library⁷⁶ verwendet. Durch die Erweiterung der OIP können okstraOWL-Instanz-Dateien als Turtle- und RDF/XML-Dateien geschrieben werden (siehe Abbildung 54). Der Export ist im Menü `File -> Export... -> OKSTRA OWL (*.rdf, *.ttl)` zu finden bzw. in der deutschen Sprachversion unter `Datei -> Exportieren... -> OKSTRA OWL (*.rdf, *.ttl)`. Es können alle OKSTRA-Versionen gelesen werden, die durch OKLABI unterstützt werden, d. h. prinzipiell alle OKSTRA-Versionen. Beim Export als OWL-Datei dient jedoch immer das OKSTRA-Schema 2.017 als Grundlage. D. h. wird beispielsweise ein Höhenplan im OKSTRA-Format 1.014 importiert, so kann dieser z. B. das Fachobjekt *Tangentenfolge* enthalten, welches vom OKSTRA 2.017 Standard aber nicht mehr unterstützt wird. In diesem Fall wird ein automatisches Upgrade der alten OKSTRA-Version zu der neuen OKSTRA-Version durchgeführt und alte Fachobjekte entsprechend durch neue ersetzt.

⁷² <https://www.cms.bgu.tum.de/oip>

⁷³ <https://bitbucket.org/tumcms/openinfraplatform>

⁷⁴ <https://github.com/tumcms/OpenInfraPlatform>

⁷⁵ <http://www.okstra.de/>

⁷⁶ <http://librdf.org/raptor/>

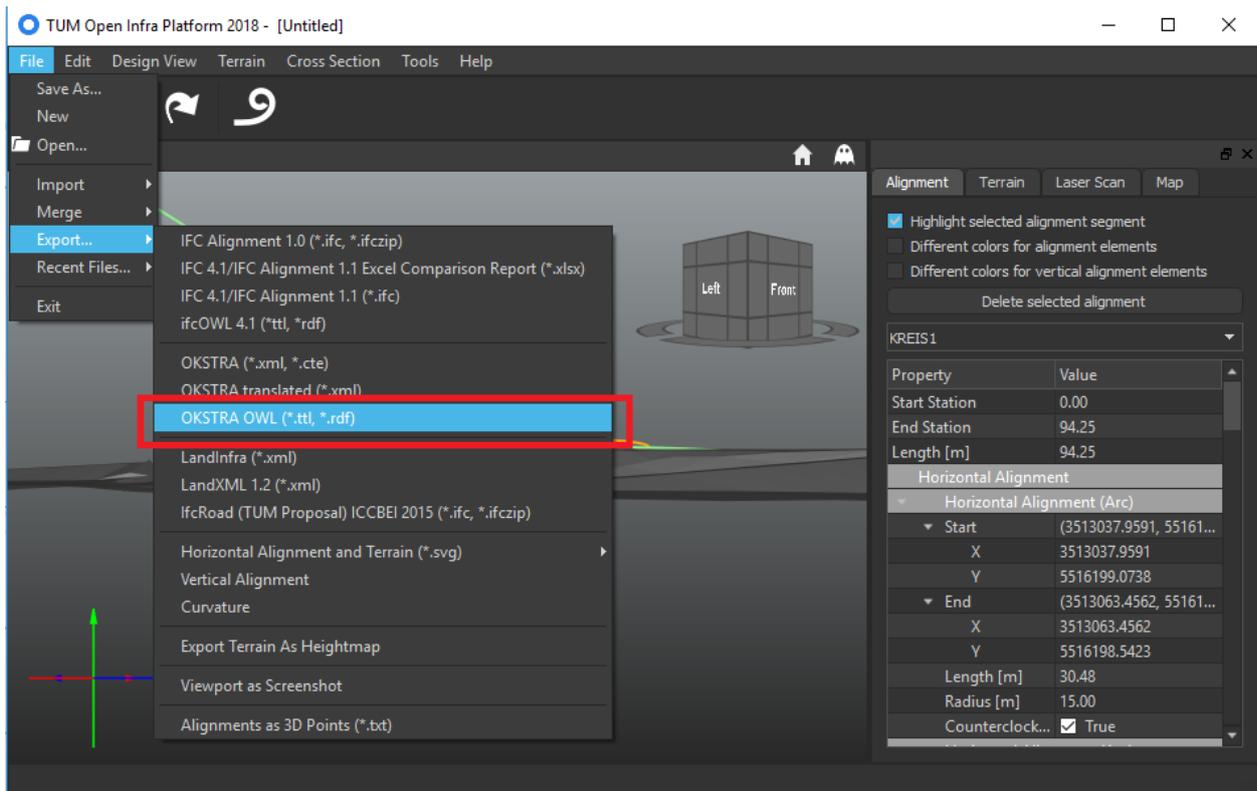


Abbildung 54: Die Open Infra Platfrom kann OkstraOWL-Dateien in der Tutrle- und RDF-Notation erstellen

Auf der OKSTRA-Webseite werden Testdaten angeboten. Unter anderem der Testdatensatz Entwurf_S_Achse-1_1014⁷⁷. Dieser wurde mithilfe der Implementierung in der Open Infra Platform als okstra-OWL-Instanzdatei im Turtle-Format exportiert. Ein Auszug daraus ist in Listing 16 dargestellt:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix okstra: <http://okstraowl.org/model/2017/okstraowl#> .

_:axis_0
  okstra:Art_Achse.1 <Hauptachse> ;
  okstra:Bezeichnung "KREIS1" ;
  okstra:hat_Achselement__Achse [
    okstra:Anfangsstation_rechnerisch__Achselement "0.000000" ;
    okstra:Elementtyp__Achselement <2> ;
    okstra:Laenge__Achselement "30.484557" ;
    okstra:Radius_am_Ende__Achselement "15.000040" ;
    okstra:Radius_zu_Beginn__Achselement "15.000040" ;
    okstra:Richtung__Achselement "6.262343" ;
    okstra:beginnt_bei_Achshauptpunkt__Achselement [
      okstra:Punktgeometrie__allgemeines_Punktobjekt "Point(3513037.96 5516199.07)" ;
      a okstra:Achshauptpunkt
    ] ;
    okstra:endet_bei_Achshauptpunkt__Achselement [
      okstra:Punktgeometrie__allgemeines_Punktobjekt "Point(3513063.46 5516198.54)" ;
```

⁷⁷ http://www.okstra.de/testdaten/geometrie_s_text_1014.zip

```

a okstra:Achshauptpunkt
];
a okstra:Achselement
], [ ...
    
```

Listing 176 Ausschnitt einer okstraOWL-Instanzdatei im Turtle-Format, die aus der OIP exportiert wurde

Abbildung 55 zeigt den RDF-Graph zum Listing 16.

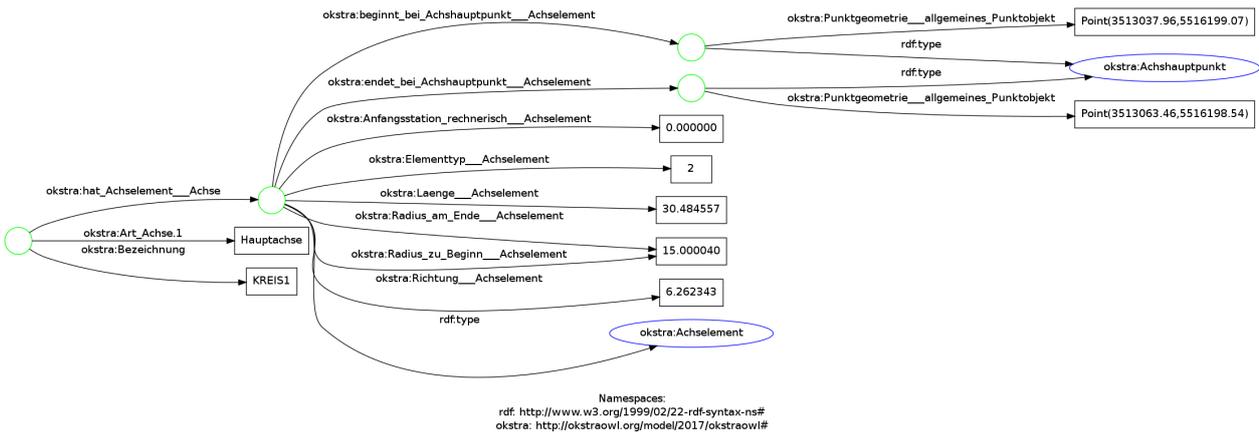


Abbildung 55: Visualisierung als RDF-Graph

Der Mehrwert dieser OWL-Repräsentation liegt darin, dass damit die Verknüpfung mit anderen Modellen auf Basis von Linked-Data- Ansätzen ermöglicht wird. Beispielsweise kann dadurch eine Verknüpfung der okstraOWL-Daten mit IFC-Daten der Version 4.1 im OWL-Format (ifcOWL 4.1) erfolgen. Ein Beispiel für ifcOWL 4.1-Daten ist in Abbildung 56 dargestellt.

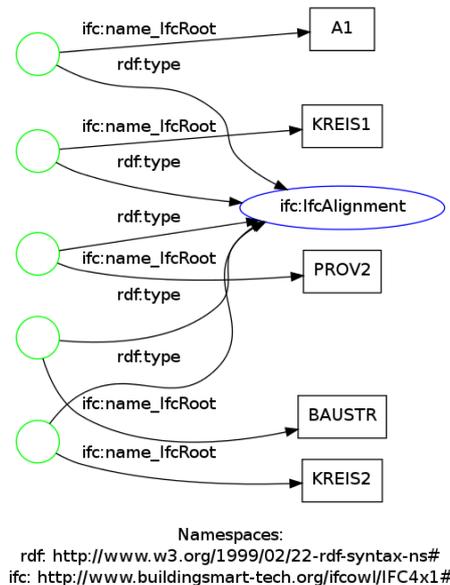


Abbildung 56: RDF-Graph von ifcOWL-Daten

Durch geeignete RDF-Abfragen (z. B. mittels SPARQL) kann dadurch herausgefunden werden, dass die Hauptachse „KREIS1“ im OKSTRA-Datensatz mit der IFC-Alignment-Instanz „KREIS1“ im ifcOWL-Datensatz korrespondiert. Als Matching-Kriterium kann man hier z. B. den Namen, die Startstationierung und den Startpunkt des Trassierungselementes wählen, um automatisiert diese Verknüpfung herzustellen. Ein anderer Weg diese Verknüpfung herzustellen, führt über die CB-NL. Diese bietet das Konzept *Middenas* an, was einer Achse in OKSTRA bzw. einen *Alignment* in der IFC-Welt entspricht. Werden beide Klassen entsprechend mit diesem Konzept markiert, erleichtert dies die Verknüpfung, da damit klargestellt wird, dass es sich hierbei um das gleiche Konzept handelt, natürlich mit unterschiedlichen Detaildarstellungen.

Denkbar wäre, dass man auf Basis dieser Verknüpfung die Datenstände abgleicht oder entsprechenden Daten ermittelt, die nur in einem der beiden Modelle vorliegen. Im Folgenden sollen kurz drei mögliche Anwendungsfälle skizziert werden:

- A) Auf OKSTRA-Seite liegt nur eine Trasse vor, jedoch kein Geländemodell. Durch die Verknüpfung kann man überprüfen, ob auf IFC-Seite ein Geländemodell zur entsprechenden Achse vorliegt. Ist diese der Fall, kann man das OKSTRA-Modell entsprechend ergänzen bzw. kann eine Fachapplikation die entsprechenden IFC-Geländedaten verwenden.
- B) Der Höhenplan von OKSTRA basiert auf Visierpunkten. Der Höhenplan von IFC 4.1 ist hingegen segmentbasiert. Durch eine Verknüpfung beider Datensätze können einem Anwender beide Repräsentationen des Höhenplans angeboten werden.
- C) IFC 4.1 unterstützt bereits grundlegende Brückengeometrien (siehe Abbildung 57). Eine Verknüpfung zwischen einer OKSTRA-Trasse und einem IFC-Alignment erlaubt es, zu einer gegebenen OKSTRA-Trasse alle Brückenbauwerke samt ihrer geometrischen Beschreibung zu ermitteln. Dabei muss das OKSTRA-Modell nicht erweitert werden, sondern erhält nur auf Basis der OWL-Beschreibung und Linked-Data-Technologie Zugriff auf diese Daten.

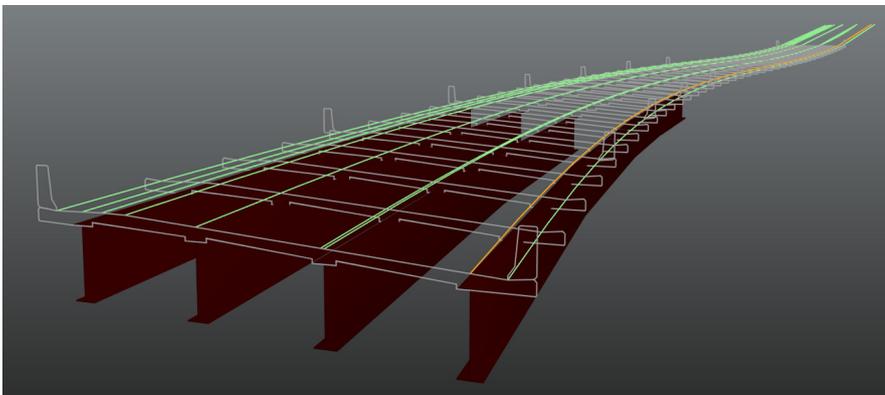


Abbildung 57: Geometrie eine Brücke im IFC-4.1-Standard

Die in den vorhergehenden Arbeitspaketen entwickelten Methoden und Technologien wurden in diesem Arbeitspaket wie zuvor beschrieben in einem Softwareprototyp umgesetzt. Dazu wurde die TUM Open Infra Plattform entsprechend erweitert, sodass eine Verknüpfung von OKSTRA/okstraOWL Objekten mit CB-NL/OTL bzw. ifcOWL Definitionen und Eigenschaften möglich ist.

Die Open Infra Plattform stellt selbst keinen SPARQL-Endpoint bereit, jedoch können die RDF-Dateien, die mittels der OIP exportiert werden einfach durch einen SPARQL-Server über einen Endpoint angeboten werden, der dann entsprechend SPARQL-Abfragen erlaubt.

3 Ergebnisse

3.1 Überblick

Im Rahmen dieses Projekts wurde eine Überführung des in UML definierten und als XML-Schema vorliegenden Datenaustauschstandards OKSTRA zur Beschreibung von Daten des Straßenwesens in eine Repräsentation auf Basis der Ontology Web Language (OWL) realisiert. Dadurch stehen nun Methoden und Techniken des Semantic Web auch für OKSTRA-Datensätze zur Verfügung. Insbesondere ist im Sinne des Linked Data Ansatzes die Verknüpfung mit Datensätzen anderer Schemata bzw. Domänen möglich. Dies kann u.a. durch Nutzung der Anfragesprache SPARQL zur integrativen Analyse der Daten der verschiedenen Ontologien erfolgen.

Wie im Zuge von Fallstudien gezeigt, können auf diese Weise beispielsweise Daten der niederländischen Straßen-Ontologie CB-NL/RWS mit OKSTRA-Daten zusammen abgefragt und analysiert werden. Dadurch lassen sich eine Reihe von grenzüberschreitenden Anwendungsszenarien realisieren, wie z.B. die Planung von Schwerlasttransporten. Andere Anwendungen von Linked Data im Straßenwesen liegen in der integrierten Analyse von 3D-Stadtmodellen im Format CityGML mit OKSTRA-Daten oder der Verknüpfung von Entwurfs- mit Bestands- bzw. Zustandsdaten des OKSTRA-Formats.

Für die Überführung von OKSTRA in okstraOWL standen eine Vielzahl unterschiedliche Abbildungsoptionen zur Verfügung, deren jeweiligen Vor- und Nachteile in diesem Bericht ausführlich dargelegt wurden. Bestimmte Eigenheiten des OKSTRA-Standards wie beispielsweise die sog. Fachbedeutungslisten erschweren zwar das Mapping, grundsätzlich ist aber eine Semantik-wahrende Überführung möglich.

Obgleich Linked Data Funktionalitäten mit der Definition von okstraOWL nun prinzipiell zur Verfügung stehen, hat sich im Zuge des Projekts (insbesondere bei der Arbeit mit realen Datensätzen) jedoch herausgestellt, dass die eigentliche Herausforderung bei der Verknüpfung verschiedener Ontologien in der unterschiedlichen semantischen Struktur und Granularität der verschiedenen Datenmodelle liegt. Auch Methoden des semiautomatischen Matchings anhand textueller Übereinstimmungen können hier nur bedingt unterstützend wirken. Stattdessen ist es erforderlich, dass der Nutzer der Abfragemechanismen Detailkenntnisse zur Semantik und Struktur der beteiligten Ontologien besitzt und anhand dessen und unter Berücksichtigung des zu erzielenden Abfrageergebnisses implizite bzw. manuelle Verknüpfungen herstellt.

Im Kontext dieses Projektes wurden verschiedene Szenarien erarbeitet, in denen der Einsatz von Linked Data (LD) Technologien im Bereich des Straßenwesens in Erwägung gezogen werden kann. Die Vor- und Nachteile spezifischer Herangehensweisen, etwa bei der Umsetzung des OKSTRA-Datenschemas in die Ontology Web Language (OWL) oder die Transformation von Instanzdaten sind in den einzelnen Abschnitten ausführlich vorgestellt, beschrieben und diskutiert. In den folgenden Abschnitten werden die Erkenntnisse und Empfehlungen der einzelnen Aspekte zusammengefasst und abschließend bewertet.

Der generelle Einsatz von Linked Data Technologien ist stark vom jeweiligen Anwendungsfall abhängig und sollte keinesfalls als „Allheilmittel“ betrachtet werden. Speziell die Erwartungen, die auf Grund der anfänglichen Ergebnisse der Wissenschaftsgemeinschaft in den Themenfeldern des Semantic Web erzielt und präsentiert wurden, wecken oft unrealistische Erwartungen. So ist der Einsatz formaler Beschreibungslogiken (engl. Description Logics), also der oberen Ebenen des sog. „Semantic Web Stacks“ – Logics, Proof, Trust – ein äußerst komplexer Bereich, der durch seine hohen Anforderungen an die Modellierung nur sehr eingeschränkt für praktische Ingenieurs- und Verwaltungsaufgaben im Bereich des Straßenwesens eingesetzt werden kann.

3.2 Wann sollte LD angewendet werden? Was kann mit LD erreicht werden?

Prinzipiell ist der Einsatz von Linked Data oft dort sinnvoll, wo eine Vielzahl heterogener Daten und Informationen gemeinsam betrachtet und verarbeitet werden müssen. Linked Data ist vor allem ein konzeptioneller und technologischer Ansatz zur Verbesserung der Interoperabilität verschiedener Informationssysteme. Dies ist im Bereich des Straßenwesens im Allgemeinen und für den Umgang mit OKSTRA-Daten

im Speziellen beispielsweise dort der Fall, wo Daten aus anderen Kontexten – bspw. dem niederländischen Straßenwesen, oder der detaillierten, flächendeckenden Modellierung von Hochbaustrukturen – in Anwendungsszenarien mit bundesdeutschen Straßendaten zusammengeführt werden müssen. Dies kann Daten betreffen, die demselben Domänenkontext entspringen (Verkehrsflüsse), aber unterschiedliche geographische Räume (z.B. Niederlande und Deutschland) betreffen und in verschiedenen Originärformaten vorliegen. Linked Data eignet sich damit auch für die Zusammenführung unterschiedlichster Infrastrukturdaten auf Europäischer oder globaler Ebene. Linked Data kann aber auch für die domänenübergreifende Analyse von Informationen eingesetzt werden, die denselben Ort im Sinne eines geographischen Umgriffs zugeordnet werden können.

Wie in den Arbeitspaketen 5 und 6 beschrieben, eignet sich Linked Data, und hier vor allem das Resource Description Framework (RDF), als neutrales Zwischenformat, um vorhandenen Datenstrukturen auf eine gemeinsame syntaktische Ebene zu bringen und auf Basis ihrer Bedeutungen (Semantik) miteinander zu verknüpfen. Anders als in anderen Interoperabilitätsformaten wie etwa der eXtended Markup Language (XML) können Verknüpfungen mithilfe von RDF sehr feingranular vorgenommen und große Datensätze effizient verarbeitet werden: Während bei XML-basierten Datenmodellen oft der gesamte Dokumentenbaum betrachtet wird, der innerhalb eines einzigen Systems vorliegen muss, können RDF-Datensätze auf der kleinsten „atomaren“ Ebene der Tripels über Systemgrenzen hinweg verteilt modelliert und verarbeitet werden.

Es soll jedoch auch betont werden, dass für das tatsächliche Linking, also die Verbindung semantisch äquivalenter Informationseinheiten, nur eingeschränkt automatisierte Verfahren (wie das Semantic Alignment) zum Einsatz kommen können. Diese liegt in der häufig stark heterogenen Struktur gewachsener Datenmodelle der unterschiedlichen Domänen bzw. Länder. Eine genaue Kenntnis der Datenmodelle, ihrer Struktur und der Semantik einzelner Entitäten ist daher unabdingbar, um sinnvolle integrierte Analyseergebnisse produzieren zu können. Es hat sich zudem gezeigt, dass die untersuchten Datenmodelle OKSTRA und RWS-OTL so verschieden aufgebaut sind, dass der Einsatz einer expliziten Verknüpfung auf Schemaebene mithilfe der starken Äquivalenzbeziehung *owl:SameAs* nicht möglich ist. Stattdessen empfehlen die Autoren die implizite Verknüpfung im Zuge entsprechender datenmodellübergreifender Anfragen, bei denen Äquivalenzbeziehungen vom Nutzer temporär und bezogen auf den unmittelbaren Kontext festgelegt werden.

Dabei liegt eine der Vorteile des Ansatzes von RDF und Linked Data in der universellen Identifizierbarkeit einzelner Bestandteile von Informationen mithilfe der Uniform Resource Identifiers (URIs). Diese Identifikatoren erlauben es, gezielt einzelne Konzepte (bspw. „Abschnitt“) oder Instanzdaten (bspw. „Linke Fahrbahn der Abfahrt 3 in westlicher Richtung der A4“) mit anderen Informationseinheiten (Konzepte der niederländischen OTL, Instanzen aus dem geographischen Stadtmodell) in Beziehung zu setzen.

Der durchgängige Einsatz von Beschreibungsformaten (RDF) und Abfragesprachen (SPARQL) sowohl für Schema- als auch für Instanzdaten über alle Fachmodelle hinweg stellt hierbei einen bedeutenden Vorteil gegenüber anderen Ansätzen wie bspw. Programmierschnittstellen (APIs, Webservices etc.) dar. In anderen heterogenen Informationsverbänden müssen meist format-, syntax- und strukturspezifische Adapter, Konverter und Schnittstellen für die Ursprungssysteme erstellt werden, die sich nicht einheitlich verarbeiten lassen. Dagegen kann die Bereitstellung vorhandener Information in Form von RDF einheitlich erfolgen und mithilfe von Triplestores/Graphendatenbanken mit universell standardisierten Abfrageschnittstellen (SPARQL) effizient verarbeitet werden, ohne sich mit den jeweiligen Systemdetails kleinteilig auseinander setzen zu müssen.

Für die oben beschriebene implizite Verknüpfung bietet sich für die betrachtete Klasse von Datenmodellen mit geographischem Bezug insbesondere der Einsatz der Sprache GeoSPARQL an, die Funktionalitäten für die Nutzung räumlicher Beziehungen (Topologie, Abstand, etc.) als Filterkriterium bereitstellt. Der geographische Kontext kann in diesem Zusammenhang einen gemeinsamen Kontext und damit eine äußerst geeignete Grundlage für das Linking verschiedener Datensätze im selben geographischen Raum bieten.

Dabei sollte jedoch berücksichtigt werden, dass sich der Aufwand z.B. für Entwickler und Anwender teilweise verlagert, weil die Auseinandersetzung etwa mit dem zugrundeliegenden Datenmodell statt mit einer Abstraktion (Schnittstelle) erfolgen muss. Einheitliche Modellierungsrichtlinien bei der Konvertierung und

Bereitstellung von Informationen als Linked Data, wie sie u.a. im „Modelling und Linking Guide“ der buildingSMART Organisation beschrieben werden, schaffen hier Abhilfe.

3.3 Unter welchen Randbedingungen ist der Einsatz von Linked Data Technologien sinnvoll?

Der Einsatz von Linked Data ist dort sinnvoll, wo verschiedene existierende Informationssysteme miteinander verknüpft oder Daten in größerem Umfang anderen zur Verfügung gestellt werden sollen. Innerhalb einzelner Abteilungen und Straßenverwaltungen mit über Jahrzehnte gewachsenen Prozessen und hochspezialisierten Werkzeugen lohnt sich der Einsatz nur dann, wenn zukünftig größere Informationskontexte in Betracht gezogen werden sollen.

Für einen Einsatz solcher Technologien und Verfahren muss die Verknüpfung mit anderen Informationen einen deutlichen Mehrwert haben, der sich aus den entsprechenden Anforderungen der jeweiligen Anwendungsfälle ergibt. Für das OKSTRA-Modell im Allgemeinen besteht ein solcher Anwendungsfall auch in der Verbreitung und Etablierung als Datenstandard innerhalb des europäischen Raums und darüber hinaus. In den derzeitigen Initiativen zur Erschaffung von weltweit einheitlichen Informationsmodellen für Infrastrukturbauten kann ein als interoperables Modell aufbereiteter und publizierter OKSTRA einen attraktiven Kandidaten für andere nationale Infrastruktursektoren darstellen. Eine Internationalisierung einschließlich der Übersetzung des Standards in andere Sprachen sowie die Integration in vorhandene Informationssysteme kann mit einem auf Linked Data basierten Schema- und Instanzmodell leichter gelingen als mit den derzeit verfügbaren technischen Voraussetzungen:

Eine wichtige Grundlage hierfür bildet die Möglichkeit, Mehrsprachigkeit durch eingebaute, etablierte RDFS-Labels zu realisieren. An die Stelle der Klassen- und Attributnamen im UML-Modell als einzige Informationsquelle (mit den entsprechenden unerwünschten Seiteneffekten bei einer Übersetzung z.B. ins Englische, bei denen alle vorhandenen Werkzeuge tiefgreifende Eingriffe im Quellcode erfordern) treten im Linked Data Ansatz die „Ressourcen“, die zwar ebenfalls natürlich-sprachliche Bezeichner haben können, aber darüber hinaus eine eindeutige Identität besitzen und durch ihre Auszeichnung mit beliebig vielsprachigen Labels sprachunabhängig verarbeitet werden können.

Die durchgehende, ggf. europaweite Nutzung von Straßeninformationssystemen erfordert daher trotz der Verfügbarkeit der Semantic Web und Linked Data Technologien eine prinzipielle Harmonisierung der Datenstrukturen insbesondere in Hinblick auf die semantische Struktur und Granularität.

Für die Einbeziehung anderer Informationssysteme auf Landes- und Bundesebene ist ein Linked Data Ansatz hilfreich, um größere, maßgeschneiderte Implementierungen zu vermeiden: Statt Schnittstellen für Abfragen von Datensätzen, z.B. über Webservices o.ä. manuell zu programmieren, zu dokumentieren und zu kommunizieren, kann ein lesender Zugriff auf OKSTRA-Instanzmodelle in Form eines SPARQL-Endpoints angeboten werden, der sich dann mit einer großen Zahl vorhandener Werkzeuge einbinden lässt.

Informationen auf Schema- und Instanzebenen können mit einem Linked Data Ansatz flexibel und leicht modularisiert werden und damit als zweckgerichteter Filter zur Reduktion von Komplexität auf Metamodell- und Modellebene eingesetzt werden: Das Konzept von Graphen, die über Systemgrenzen hinaus miteinander verknüpft werden können, lässt sich mit den W3C Sprachstandards leichter einheitlich umsetzen als mit herkömmlichen Modellierungsansätzen auf Basis von EXPRESS oder UML/XML.

Ein weiteres Beispiel für einen möglichen sinnvollen Einsatz ist die Zusammenführung von Entwurfs- und Bestands-, sowie Zustandsdaten die derzeit in verschiedenen Kontexten verarbeitet werden. Eine endgültige Empfehlung hängt aber auch von einer detaillierten Analyse der vorhandenen Werkzeuge, der in der Realität vorhandenen Daten und dem Mehrwert der Verknüpfungen ab.

Für große Datenmengen, wie sie bei der BAST anfallen, ist die Skalierbarkeit von generischen Graphdatenbanken oder Triplestores gegenüber anderen hochspezialisierten Werkzeugen (z.B. relationalen Datenbanken) ein weitergehend zu untersuchender Punkt, der entscheidend für den breiten Einsatz von Linked Data ist. Obwohl es zahlreiche Beispiele für große Datensätze mit effizienten Abfrageleistungen gibt⁷⁸, ist

⁷⁸ eine Übersicht findet sich etwa auf <https://www.w3.org/wiki/LargeTripleStores>

deren Einsatz stark von den jeweiligen Anforderungen, den Größen der entsprechenden Datensätze und der Komplexität der Abfragen abhängig.

Eine weitere Einschränkung beim Umgang mit OKSTRA-Daten ist die Komplexität des Modells, die durch die zugrundeliegenden Modellierungsprinzipien von Linked Data (Triples, Listen, OWA etc.) weiter erhöht wird. Die Effizienz des Umganges mit Linked Data ist stark von der Formulierung der Abfragen abhängig. Der bislang eher niedrige Verbreitungsgrad von Linked Data Technologien in der IT-Industrie (die jedoch teilweise auch für andere Technologien im Zusammenhang mit OKSTRA zutrifft) ist eine weitere Hürde, die den effizienten Einsatz erschwert.

Literatur

- Allemang, D., Hendler, J., 2011. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Elsevier.
- Bakker, J., Mommers, B., 2016. A hybrid linked data approach to support asset management, in: *Life-Cycle of Engineering Systems, Life-Cycle of Civil Engineering Systems*. CRC Press, pp. 126–126. doi:10.1201/9781315375175-79
- Borrmann, André; Beetz, Jakob; Koch, Christian; Liebich, Thomas: *Industry Foundation Classes – Ein herstellerunabhängiges Datenmodell für den gesamten Lebenszyklus eines Bauwerks*. In: Borrmann, André; König, Markus; Koch, Christian; Beetz, Jakob (Eds.): *Building Information Modeling*, Springer Fachmedien Wiesbaden, 2015
- Beetz, J., van Leeuwen, J., de Vries, B., 2009. IfcOWL: A Case of Transforming EXPRESS Schemas into Ontologies. *AI EDAM* 23, 89–101. doi:10.1017/S0890060409000122
- Beetz, J., van Leeuwen, J.P., de Vries, B., 2005. An Ontology Web Language Notation of the Industry Foundation Classes, in: Scherer, R.J., S.-E (Eds.), *Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction, CIB-W78*. Presented at the CIB W78, Dresden, Germany, pp. 193–198.
- Berners-Lee, T., Hendler, J., Lassila, O., 2001. The semantic web. *Sci. Am.* 284, 28–37.
- Bizer, C., Heath, T., Berners-Lee, T., 2009. *Linked Data - The Story So Far*. *Int. J. Semantic Web Inf. Syst.* 5, 1–22. doi:10.4018/jswis.2009081901
- Drummond, N., Rector, A.L., Stevens, R., G, M., Horridge, M., Wang, H., Seidenberg, J., 2006. Putting OWL in Order: Patterns for Sequences in OWL, in: *Proceedings of 2nd OWL Experiences and Directions Workshop*, Athens.
- Giertz, L.M., 1982. *SFB and Its Development 1950-1980*. CIB/SFB International Bureau on behalf of the author; Foras Forbartha distributor, Dublin, Ireland.
- Kolbe, T. H., Gröger, G., Plümer, L., 2005. "CityGML: Interoperable access to 3D city models." *Geo-information for disaster management*. Springer, Berlin, Heidelberg, 2005. 883-899.
- Krima, S., Barbau, R., Fiorentini, X., Sudarsan, R., Sriram, R.D., 2009. *Ontostep: Owl-dl ontology for step*. Natl. Institut Stand. Technol. NISTIR 7561.
- Pauwels, P., Terkaj, W., 2016. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Autom. Constr.* 63, 100–133. doi:10.1016/j.autcon.2015.12.003
- Pauwels, P., Terkaj, W., Krijnen, T., Beetz, J., 2015. Coping with lists in the ifcOWL ontology, in: *22nd EG-ICE International Workshop*. pp. 113–122.
- Schevers, H., Drogemuller, R., 2005. Converting the Industry Foundation Classes to the Web Ontology Language, in: *Semantics, Knowledge and Grid, 2005. SKG'05. First International Conference on*. p. 73–73.
- RWS 2017: *Instructie modelleren op basis van de OTL 2.1*, Rijkswaaterstaat, 2017
- Stoter, J., Beetz, J., Ledoux, H., Reuvers, M., Klooster, R., Janssen, P., Penninga, F., Zlatanova, S., Brink, L. van den, 2013. Implementation of a National 3D Standard: Case of the Netherlands, in: Pouliot, J., Daniel, S., Hubert, F., Zamyadi, A. (Eds.), *Progress and New Trends in 3D Geoinformation Sciences, Lecture Notes in Geoinformation and Cartography*. Springer Berlin Heidelberg, pp. 277–298.