

DIN EN 17632



ICS 35.240.67

Einsprüche bis 2021-03-29

Entwurf**Semantischer Modellierungs- und Verknüpfungsstandard (SMLS) für die Datenintegration in der gebauten Umwelt;
Deutsche und Englische Fassung prEN 17632:2021**

Building Information Modelling (BIM) –
Semantic Modelling and Linking (SML);
German and English version prEN 17632:2021

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2021-01-29 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfs besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an nabau@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an den DIN-Normenausschuss Bauwesen (NABau), 10772 Berlin oder Saatwinkler Damm 42/43, 13627 Berlin.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 291 Seiten

DIN-Normenausschuss Bauwesen (NABau)

Nationales Vorwort

Dieses Dokument (prEN 17632:2021) wurde vom Technischen Komitee CEN/TC 422 „Building Information Modelling (BIM)“ erarbeitet, dessen Sekretariat von SN (Norwegen) gehalten wird.

Das zuständige deutsche Normungsgremium ist der Arbeitsausschuss NA 005-13-04 AA „Datenstrukturen für BIM-Kataloge (SpA zu CEN/TC 442/WG 4, ISO/TC 59/SC 13/WG 6)“ im DIN-Normenausschuss Bauwesen (NABau).

Um Zweifelsfälle in der Übersetzung auszuschließen, ist die englische Originalfassung beigelegt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen Text.

Für die in diesem Dokument zitierten Normen wird im Folgenden auf die entsprechenden deutschen Dokumente hingewiesen:

ISO 11354-1	siehe	DIN EN ISO 11354-1
ISO 16739-1	siehe	DIN EN ISO 16739-1 (z. Z. Entwurf)
ISO 19650-1	siehe	DIN EN ISO 19650-1
ISO 19650-2	siehe	DIN EN ISO 19650-2
ISO 19650-3	siehe	DIN EN ISO 19650-3 (z. Z. Entwurf)
ISO 19650-5	siehe	DIN EN ISO 19650-5 (z. Z. Entwurf)

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Nationaler Anhang NA (informativ)

Literaturhinweise

DIN EN ISO 11354-1, *Fortgeschrittene Automatisierungstechnologien und deren Anwendung — Anforderungen für das Erreichen einer Prozessinteroperabilität in Fertigungsunternehmen — Teil 1: Rahmenwerk für die Unternehmensinteroperabilität*

DIN EN ISO 16739-1, *Industry Foundation Classes (IFC) für den Datenaustausch in der Bauwirtschaft und im Anlagenmanagement — Teil 1: Datenschema*

DIN EN ISO 19650-1, *Organisation und Digitalisierung von Informationen zu Bauwerken und Ingenieurleistungen, einschließlich Bauwerksinformationsmodellierung (BIM) — Informationsmanagement mit BIM — Teil 1: Begriffe und Grundsätze*

DIN EN ISO 19650-2, *Organisation und Digitalisierung von Informationen zu Bauwerken und Ingenieurleistungen, einschließlich Bauwerksinformationsmodellierung (BIM) — Informationsmanagement mit BIM — Teil 2: Planungs-, Bau- und Inbetriebnahmephase*

DIN EN ISO 19650-3, *Organisation von Informationen zu Bauwerken — Informationsmanagement mit Bauwerksinformationsmodellierung — Teil 3: Betriebsphase der Assets*

DIN EN ISO 19650-5, *Organisation von Daten zu Bauwerken — Informationsmanagement mit BIM — Teil 5: Spezifikation für Sicherheitsbelange von BIM, der digitalisierten Bauwerke und des smarten Assetmanagements*

— Leerseite —

Titel de: Semantischer Modellierungs- und Verknüpfungsstandard (SMLS) für die Datenintegration in der gebauten Umwelt

Titel en: Building Information Modelling (BIM) — Semantic Modelling and Linking (SML)

Titel fr:

Inhalt

	Seite
Europäisches Vorwort	4
Einleitung	5
1 Anwendungsbereich	7
2 Normative Verweisungen	8
3 Begriffe	9
4 Symbole und Abkürzungen	12
4.1 Symbole	12
4.2 Abkürzungen	12
5 Fähigkeitsniveaus	14
6 Konzeptionelle L1: Datensprache	15
7 Konkrete L1: Datensprachbindungen	18
7.1 Einleitung	18
7.2 Identifizierung: URI-Strategie	21
7.3 Namenskonventionen	23
7.4 Anmerkung	23
7.5 Aufzählungsdatentypen	25
7.6 Zergliederung (Instanz-Ebene)	25
7.7 Arten und Einheiten von Größen	25
7.8 Größenmodellierung	25
7.9 Gruppierung	26
8 Generisches M1: Datenmodell der obersten Ebene	27
8.1 Modell der obersten Ebene	27
8.2 Systems-Engineering-Erweiterung	31
8.2.1 Einleitung	31
8.2.2 Konformität mit Anforderungen	32
9 Implementierung von SML in Code	33
10 Vernetzung von Daten	33
10.1 Arten von Vernetzungen	33
10.2 Sprachvernetzungsätze auf Sprachebene	34
11 Konformität	34
11.1 Allgemeines	34
11.2 Konformität auf Sprachebene	35
11.3 Konformität auf semantischer Ebene	35
Anhang A (normativ) Konzeptionelle Datensprache und Datenmodell der obersten Ebene	36
A.1 Allgemeines	36
A.2 in SKOS (Turtle-Format)	36
A.3 in RDFS (Turtle-Format)	41
A.4 in OWL (Turtle-Format)	49
A.5 in SHACL (Turtle-Format)	59
Anhang B (normativ) Ausgewählte W3C RDF Sprachteilsätze	72
B.1 Allgemeines	72

B.2	XML-Schema Teil 2: Datentypen 2. Ausgabe.....	72
B.3	Resource Description Framework (RDF)	72
B.4	Simple Knowledge Organization System (SKOS)	73
B.5	Resource Description Framework Schema (RDFS).....	73
B.6	Web Ontology Language (OWL)	74
B.7	Shape Constraint Language (SHACL).....	75
Anhang C (informativ) Beispiel in SKOS.....		78
C.1	Allgemeines	78
C.2	Beispiel in SKOS (RDF/XML-Format)	78
C.3	Beispiel in SKOS (Turtle-Format).....	79
C.4	Beispiel in SKOS (JSON-LD-Format).....	80
Anhang D (informativ) Beispiel in RDFS.....		83
D.1	Allgemeines	83
D.2	Beispiel in RDFS (RDF/XML-Format)	83
D.3	Beispiel in RDFS (Turtle-Format).....	85
D.4	Beispiel in RDFS (JSON-LD-Format).....	88
Anhang E (informativ) Beispiel in OWL		93
E.1	Allgemeines	93
E.2	Beispiel in OWL (RDF/XML-Format)	93
E.3	Beispiel in OWL (Turtle-Format).....	95
E.4	Beispiel in OWL (JSON-LD-Format).....	98
Anhang F (informativ) Beispiel in SHACL.....		104
F.1	Allgemeines	104
F.2	Beispiel in SHACL (RDF/XML-Format)	104
F.3	Beispiel in SHACL (Turtle-Format)	107
F.4	Beispiel in SHACL (JSON-LD-Format)	110
Anhang G (informativ) Zusammenhang mit anderen Asset-/Produktmodellierungsnormen.....		116
G.1	Allgemeines	116
G.2	Zusammenhang mit ISO 21597	116
G.3	Zusammenhang mit ISO 23387	117
G.3.1	Einleitung	117
G.3.2	Modellierungszusammenhang mit ISO 12006-3.....	117
G.3.3	Beispiel CRB.....	118
G.3.4	Vollständiges CRB-Code-Beispiel in SHACL (im Turtle-Format).....	127
Literaturhinweise.....		132

Europäisches Vorwort

Dieses Dokument (prEN 17632:2021) wurde vom Technischen Komitee CEN/TC 422 „Building Information Modelling (BIM)“ erarbeitet, dessen Sekretariat von SN gehalten wird.

Dieses Dokument ist derzeit zur CEN-Umfrage vorgelegt.

Einleitung

Der Kontext dieses Dokuments ist die gebaute Umwelt. Die Assets von Bauwerken oder Infrastruktur müssen während ihres gesamten Lebenszyklus einschließlich der Programmierung, Gestaltung, des Baus und des Betriebs (wie in der Normenreihe ISO 19650 definiert) und in der zu ihrer Herstellung und Lieferung verwendeten Lieferkette verwaltet werden. Riesige Datenmengen zu diesen Assets werden erzeugt, in zahlreichen unterschiedlichen Formaten und Datenstrukturen weitergegeben und gehen oftmals wieder verloren. Um die Assets effizient und effektiv entsprechend den gebräuchlichen Normen im Asset-Management (wie durch die Normenreihe ISO 55000 definiert) zu verwalten, müssen die Daten auffindbar, zugänglich, interoperabel und wiederverwendbar sein (FAIR).¹

Das World Wide Web Consortium (W3C) bietet sogenannte vernetzte Daten (en: linked data, LD) und Technologien für das semantische Web (en: semantic web, SW) [1], die in der Lage sind, Daten eine allgemeine Form (Syntax) und Bedeutung (Semantik) zu verleihen und Daten so in einer lieferantenneutralen Art und Weise FAIR zu machen.

Ziel dieses Dokuments ist es, die Anwendung dieser Technologie für die gebaute Umwelt zu normen, um Daten zu ermöglichen, die dem FAIR-Prinzip entsprechen. Dieses Dokument legt fest, wie die Bau- und Softwareindustrie diese Technologie der vernetzten Daten und des semantischen Webs anwenden.

Hierdurch folgt es dem Prinzip der Wahrung einer möglichst einfachen und genormten semantischen Modellierung (Tabelle 1).

Tabelle 1 — Streben nach genormter und einfacher Semantik

Semantik	Norm	Proprietär
Einfach	OK	SO FERN ERFORDERLICH
Komplex	SO FERN ERFORDER- LICH	NICHT OK

Mit anderen Worten: Es ist nicht Absicht dieses Dokuments, irgendjemanden davon zu überzeugen, bereits vorhandene Datenstrukturen aufzugeben. Vielmehr schlägt dieses Dokument vor, diese Daten in einer auffindbaren, zugänglichen, interoperablen und wiederverwendbaren Weise (FAIR) zu speichern, zu modellieren, zu veröffentlichen und zu vernetzen. Dies dient dem Nutzen der Industrie von der Planung und Gestaltung bis zum Bau und Betrieb.

Dieses Dokument ergänzt andere ISO-Normen ohne Überschneidung. Die verbundenen ISO-Normen sind im Anhang G mit einer genauen Beschreibung ihrer Beziehung aufgelistet.

Die Anwendung dieses Dokuments auf neue oder bestehende Software wird zu einer zukunftssicheren semantischen Dateninteroperabilität führen, d.h. einer Interoperabilität, durch die die Bedeutung des Datenmodells innerhalb des Kontexts des Themenbereichs von den teilnehmenden Systemen verstanden wird [QUELLE: ISO/IEC 19941].

¹ Ein übliches Prinzip von go-fair.org im modernen Datenmanagement.

Darüber hinaus werden die Datensätze und Datenmodelle kompatibel, wiederverwendbar, kombinierbar und somit integral anwendbar. Dieses Dokument ermöglicht Entscheidungsfindungen über die gesamte Lebenszyklusdauer und in der beteiligten Lieferkette.

Das Datenmanagement in der gebauten Umwelt ist durch drei Hauptkategorien gekennzeichnet:

- Begriffe und Definitionen von Daten (auf Objektebene und Attributebene);
- Datenaustausch (Datenübertragung von einer Partei zu einer anderen Partei) und Datenteilung (Veröffentlichung der Daten durch eine Partei, wo sie für andere Parteien zugänglich sind);
- Datenintegration einschließlich der Vernetzung aller Daten.

Für jede dieser Kategorien gelten die Interoperabilitätsansätze, wie durch das Rahmenwerk für die Unternehmensinteroperabilität (en: enterprise interoperability framework, EIF) (ISO 11354-1) definiert:

- 1) *vereinheitlichte Vorgehensweise*, mit einigen *gemeinsamen Meta-Modellen*;
- 2) *integrierter Ansatz* mit einigen *gemeinsamen Formen*;
- 3) *Vereinigungsvorgehensweise*, keine gemeinsamen Formen oder Meta-Modelle, sondern dynamische Anpassung/Einstellung.

Diese dem EIF entsprechenden Ansätze gelten für Geschäftsprozess-, Dienst- und Datenaspekte.

1 Anwendungsbereich

Dieses Dokument behandelt einen integrierten Ansatz und eine vereinheitlichte Vorgehensweise für Datenaspekte und insbesondere für Assets und gebaute Umwelt unter Verwendung der EIF-Terminologie.

Die folgende Datenarchitektur (Bild 1) gilt innerhalb jeder Kategorie.

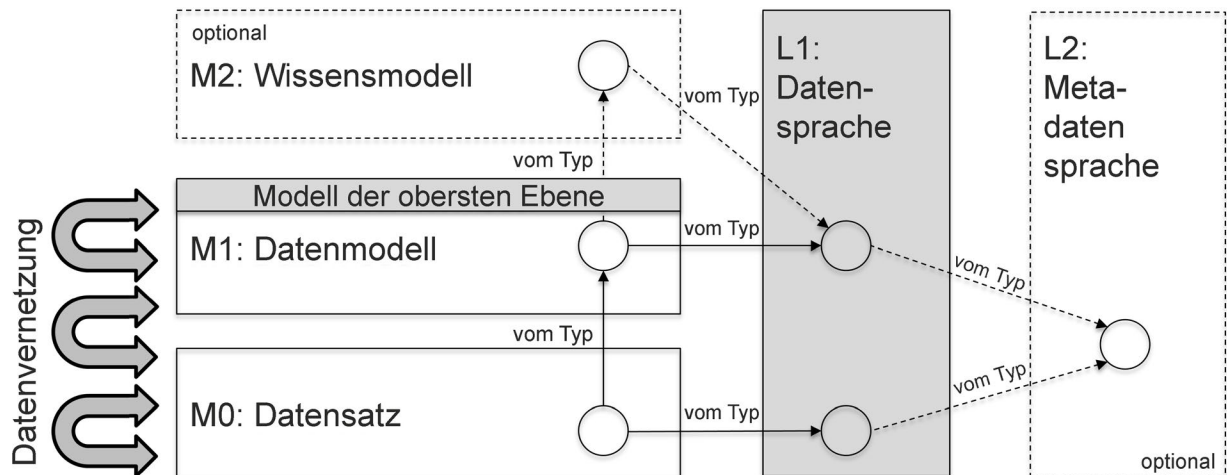


Bild 1 — Datenarchitektur mit Typologie (graue Bereiche weisen auf den Anwendungsbereich dieses Dokuments hin)

Dieses Dokument legt fest:

- eine oberste Ebene „M1: Datenmodell“ als gemeinsame Form;
- eine konzeptionelle Ebene „L1: Datensprache“ als gemeinsames Meta-Modell mit vier auf „vernetzten Daten“ basierenden konkreten Sprachbindungen (SKOS, RDFS, OWL und SHACL), einschließlich:
 - einer Auswahl RDF-basierter Formate (anzuwenden für alle Modellierungs- und Sprachebenen);
 - ein Satz von Datenmodellierungsmustern (zur Kennzeichnung, Benennung, Handhabung von Aufzählungstypen, Größenmodellierung, Asset-Zergliederung usw.).
- ein Vernetzungsansatz für die Vernetzung von Datensätzen, die Vernetzung von Datenmodellen und die Vernetzung von Datensätzen und Datenmodellen, die innerhalb der gebauten Umwelt aus verschiedenen Perspektiven relevant sind, wie etwa:
 - Bauwerksinformationsmodellierung (BIM);
 - Geoinformationssysteme (GIS);
 - Systems Engineering (SE)²;
 - Überwachung und Steuerung (M&C);
 - elektronisches Dokumentenmanagement (EDM).

2 Der interdisziplinäre Ansatz zur Steuerung der gesamten erforderlichen technischen und betrieblichen Anstrengungen zur Umwandlung einer Reihe von Stakeholder-Anforderungen, -Erwartungen und -Beschränkungen in eine Lösung und zur Unterstützung dieser Lösung während ihrer gesamten Lebensdauer [QUELLE: ISO/IEC/IEEE 12207:2017(en), 3.1.65].

Dieses Dokument legt kein Wissensmodell fest, da dies bereits in ISO 12006-3 verfügbar ist.

Dieses Dokument legt keine Metadatensprache fest, da diese bereits durch die konkreten RDF-Sprachbindungen (die RDFS) bereitgestellt ist.

Der Anwendungsbereich dieses Dokuments schließt grundsätzlich Folgendes aus:

- Modellierung von Geschäftsprozessen;
- Aspekte der Softwareimplementierung;
- Datenbündelungs- und Transport-/Transaktionsaspekte [gehandhabt durch ISO TC59/SC13 Information container for document delivery (ICDD), respektive verschiedene Initiativen in Verbindung mit dem Handbuch der Informationslieferungen (IDM)/Informationsaustausch-Anforderungen (EIR)];
- bereichsspezifische (hier: spezifisch für gebaute Umwelt) Inhaltsmodellierung in Form von Konzepten, Attributen und Relationen auf Endnutzer-Ebene (die tatsächlichen Ontologien selbst) über eine generische obere Ontologie und Modellierungsmuster hinaus.

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

JSON-LD 1.1, A JSON-based Serialization for Linked Data, W3C Candidate Recommendation, 17 April 2020, <https://www.w3.org/TR/json-ld11/>

OWL ^{2 3} Web Ontology Language, Document Overview (Second Edition), W3C Recommendation, 11 December 2012, <https://www.w3.org/TR/2012/REC-owl2-overview-20121211/>

RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation, 25 February 2014, <https://www.w3.org/TR/rdf11-concepts/>

RDF 1.1 Turtle, W3C Recommendation, 25 February 2014, <https://www.w3.org/TR/turtle/>

RDF 1.1 XML Syntax, W3C Recommendation 25 February 2014, <https://www.w3.org/TR/rdf-syntax-grammar/>

RDF Schema 1.1, W3C Recommendation, 25 February 2014, <https://www.w3.org/TR/rdf-schema/>

SHACL (Shapes Constraint Language), W3C Recommendation, 20 July 2017, <https://www.w3.org/TR/shacl/>

SKOS Simple Knowledge Organization System Reference, W3C Recommendation, 18 August 2009, <https://www.w3.org/TR/skos-reference/>

SPARQL 1.1 Overview, 21 March 2013, W3C Recommendation, <https://www.w3.org/TR/sparql11-overview/> (unter anderem mit Verweisung auf die nächsten beiden spezifischeren Verweisungen)

SPARQL 1.1 Query Language, W3C Recommendation, 21 March 2013, <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/>

3 Im weiteren Verlauf „OWL“ genannt.

SPARQL 1.1 Protocol, W3C Recommendation, 21 March 2013, <https://www.w3.org/TR/sparql11-protocol/>

XML Schema Part 2: Datatypes, Second Edition, W3C Recommendation, 28 October 2004, <https://www.w3.org/TR/xmlschema-2/>

3 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach ISO 6707-1 und die folgenden Begriffe.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- ISO Online Browsing Platform: verfügbar unter <https://www.iso.org/obp>
- IEC Electropedia: verfügbar unter <http://www.electropedia.org/>

3.1

Asset

Objekt, Sache oder Einheit, das bzw. die für eine Organisation einen potentiellen oder tatsächlichen Wert besitzt

[QUELLE: ISO 55000:2014, 3.2.1, modifiziert — die Anmerkungen 1, 2 und 3 zum Begriff wurden gestrichen.]

3.2

maschinenlesbar

von einem Computer lesbar und verarbeitbar

3.3

maschineninterpretierbar

von einem Computer semantisch interpretierbar

3.4

Fähigkeitsniveau

LoC

(en: level of capability)

Ebene der Modellierungsleistungsfähigkeit auf semantischer Ebene innerhalb eines Datenmodells zur Erfüllung des Anwendungsfalltyps

3.5

Format

vorgegebene Anordnung von Daten auf einem Datenträger

[QUELLE: ISO 5127:2017, 3.1.13.12]

3.6

Ontologie

formale ausdrückliche Spezifikation einer gemeinsamen Konzeptionalisierung

Anmerkung 1 zum Begriff: Eine Ontologie beinhaltet üblicherweise Definitionen von Konzepten und festgelegten Beziehungen zwischen diesen, die in formaler Weise dargelegt sind, sodass eine Maschine sie zur Abwägung nutzen kann.

Anmerkung 2 zum Begriff: In diesem Dokument angewendet als eine Reihe von Konzepten, (Referenz-) Einzelanfertigungen, Wertetypen, (Referenz-) Werten, Attributen, Relationen, Beschränkungen und Ableitungen.

[QUELLE: ISO 5127:2017, 3.1.2.03, modifiziert — Anmerkung 2 zum Begriff hinzugefügt]

3.7

Typologie

Hierarchie in Verbindung mit Klassifizierungs-/Instanziierungsrelationen

[QUELLE: ISO/IEC 11179-3:2013, 3.2.135, modifiziert — in der englischen Sprachfassung wurden die Wörter „type of“ gestrichen. Die Wörter „which deals with“ wurden zu „related to“ geändert. Das Wort „relationships“ wurde zu „relations“ geändert.]

3.8

Taxonomie

Hierarchie in Verbindung mit Generalisierungs-/Spezialisierungsrelationen

[QUELLE: ISO/IEC 11179-3:2013, 3.2.135, modifiziert — in der englischen Sprachfassung wurden die Wörter „type of“ gestrichen. Die Wörter „which deals with“ wurden zu „related to“ geändert. Das Wort „relationships“ wurde zu „relations“ geändert. Keine Vermischung von Konzepten, Attributen und Relationen.]

3.9

Meronomie

Hierarchie in Verbindung mit Teil-Ganz-Beziehungen (Zergliederung)

[QUELLE: ISO/IEC 11179-3:2013, 3.2.135, modifiziert — in der englischen Sprachfassung wurden die Wörter „type of“ gestrichen. Die Wörter „which deals with“ wurden zu „related to“ geändert. Das Wort „relationships“ wurde zu „relations“ geändert.]

3.10

gebaute Umwelt

Gesamtheit der von Menschen erstellten oder verursachten physischen Objekte in einem bestimmten Bereich oder Raum

[QUELLE: ISO 6707-3:2017, 3.1.3]

3.11

Tripel

Aussage in der Form „Subjekt, Prädikat, Objekt“, die eine Relation ausdrückt

3.12

Fähigkeitsniveau

Modellierungsleistungsfähigkeit in Verbindung mit den Bedürfnissen eines spezifischen Anwendungsfall-typs, bereitgestellt durch die vernetzten Datensprachen

3.13

Objekt

jeder Teil der erfassbaren und beschreibbaren Umwelt

Anmerkung 1 zum Begriff: Ein Objekt ist das abstrakte oder physische Ziel von Gedanken, Gefühlen oder Handlungen.

Anmerkung 2 zum Begriff: Innerhalb dieses Entwurfs werden die Begriffe Instanz und Individuum als Synonyme für ein Objekt verwendet.

[QUELLE: ISO 12006-2:2015, 3.1.1, modifiziert — Anmerkung 2 zum Begriff hinzugefügt.]

3.14

Konzept

abstrakte Einheit zur Bestimmung der Zugehörigkeit zu einer Kategorie

[QUELLE: ISO/IEC 2382:2015, 2122971]

3.15

Merkmal

inhärente oder erworbene Eigenschaft eines Objekts

3.16

Attribut

Datenelement für die rechnerinterpretierbare Beschreibung eines Merkmals, einer Relation oder einer Klasse

[QUELLE: ISO 22274:2013, 3.2]

3.17

Relation

Sinn, in dem Konzepte über einzelne Rollen verbunden werden können

BEISPIEL Kausalität ist eine Relation mit zwei einzelnen Rollen: Ursache und Wirkung.

[QUELLE: ISO/IEC 11179-3:2013, 3.2.119]

3.18

Datensatz

Gruppe von Dateninstanzen, die etwas festlegen oder beschreiben, auf das man in der Realität zeigen kann oder könnte

3.19

Datenmodell

Spezifikation/Beschreibung der Organisation der bedeutungsgebenden Daten (Semantik) eines Datensatzes

3.20

Austausch-Informationsanforderung

EIR

(en: exchange information requirement)

Informationsanforderung im Zusammenhang mit einer Informationsbestellung

[QUELLE: ISO 19650-1:2018 3.3.6]

3.21

Systems Engineering

SE

interdisziplinärer Ansatz, der sämtliche technischen und führungsbezogenen Anstrengungen leitet, die erforderlich sind, um eine Gemengelage aus Stakeholder-Bedürfnissen, Erwartungshaltungen und Vorgaben in eine Lösung zu überführen und diese Lösung während deren gesamten Lebensdauer zu unterstützen

[QUELLE: ISO/IEC/IEEE 12207:2017, 3.1.65]

3.22

Metadaten

Daten über Daten (Dokumente, Datensätze, Datenmodelle oder darin enthaltene Elemente)

3.23

Datenmodell der obersten Ebene

allgemeinste Taxonomie als Teil eines Datenmodells

4 Symbole und Abkürzungen

4.1 Symbole

Dieses Dokument enthält keine Symbole.

4.2 Abkürzungen

Für die Anwendung dieses Dokuments gelten die folgenden Abkürzungen.

API	Programmierschnittstelle (en: application programming interface)
BIM	Bauwerksinformationsmodellierung (en: building information modelling)
DT	Datenvorlage (en: data template) [CEN TC 442]
ECMA	European computer manufacturers association international
EDM	elektronisches Datenmanagement (en: electronic data management)
EIF	Rahmenwerk für die Unternehmensinteroperabilität (en: enterprise interoperability framework)
EIR	Austausch-Informationsanforderungen (en: exchange information requirements)
FAIR	auffindbar, zugänglich, interoperabel, wiederverwendbar (en: findable, accessible, interoperable, reusable) [go-fair.org]
FO	funktionales Objekt
GIS	Geoinformationssysteme (en: geo-spatial information systems)
GUID	globaler eindeutiger Bezeichner (en: globally unique identifier) (üblicherweise zugewiesen)
ICDD	Informationscontainer zur Datenübergabe (en: information container for linked document delivery) [ISO]
ID	Bezeichner (en: identifier)
IDM	Handbuch der Informationslieferungen (en: information delivery manual)
IFC	Industry Foundation Classes [ISO]
IETF	Internet Engineering Task Force
IO	imaginäres Objekt
JSON	JavaScript-Objektnotation (en: JavaScript object notation) [ECMA]
JSON-LD	JavaScript-Objektnotation — vernetzte Daten (en: JavaScript object notation — linked data) [W3C]
LBD CG	Linked Building Data Community Group [W3C]
LD	vernetzte Datentechnologie (en: linked data technology) [W3C]

LoC	Fähigkeitsniveau (en: level of capability)
M&C	Überwachung und Steuerung (en: monitoring & control)
OMG	Objektmanagementgruppe (en: object management group)
OWL	Web-Ontologiesprache (en: web ontology language) [W3C]
QUDT	Größen, Einheiten und Datentypen (en: quantities, units & data types) [qudt.org]
RDF	Resource Description Framework [W3C]
RDFS	Resource Description Framework Schema [W3C]
RFC	Bitte um Stellungnahme (en: request for comments) [IETF]
RO	reales Objekt
SE	Systems Engineering
SHACL	Shapes Constraints Language [W3C]
SML	semantische Modellierung und Vernetzung (en: semantic modelling and linking) [CEN]
SPARQL	SPARQL-Protokoll und RDF-Abfragesprache [W3C]
SPFF	physisches STEP-Dateiformat (en: STEP physical file format) [STEP]
STEP	Norm für den Austausch von Produktmodelldaten (en: standard for the exchange of product model data) [ISO]
SSoF	einzelne Faktenquelle (en: single source of facts)
SW	semantische Webtechnologie (en: semantic web technology) [W3C]
TO	technisches Objekt
UML	vereinheitlichte Modellierungssprache (en: Unified Modeling Language) [OMG]
URI	einheitlicher Ressourcenbezeichner (en: Uniform Resource Identifier) [W3C]
UUID	universell eindeutiger Bezeichner (en: universally unique identifier) [IETF]
XML	erweiterbare Auszeichnungssprache (en: Extensible Markup Language) [W3C]
XSD	Schema-Definition der erweiterbaren Auszeichnungssprache (en: extensible markup language schema definition) [W3C]
W3C	World Wide Web Consortium
WWW	World Wide Web [W3C]

5 Fähigkeitsniveaus

Verschiedene Anwendungsfalltypen erfordern eine Spezifikation für verschiedene Fähigkeitsniveaus (LoC) in Verbindung mit der erforderlichen Modellierungsleistungsfähigkeit. Dieses Dokument legt drei Haupt-LoC fest (Bild 2):

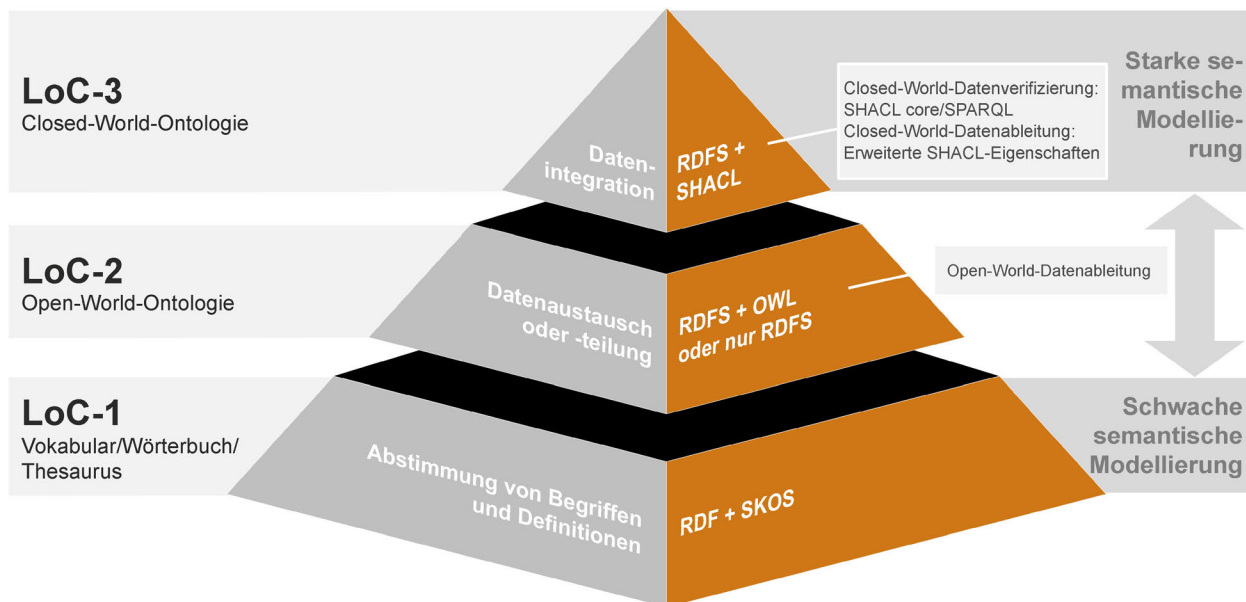


Bild 2 — Drei Haupt-Anwendungsfalltypen und damit verbundene „Fähigkeitsniveaus (LoC)“

Der linke/graue Teil des Bilds zeigt die Unternehmensseite, die rechte/orangefarbene Seite die verwendete Modellersprache. Der einfachste Anwendungsfalltyp, der die schwächste semantische Modellierung erfordert, ist das gemeinsame Verständnis und die *Abstimmung von Begriffen und Definitionen* für die Beschreibung von Assets, ihrer Umgebung und der internen Struktur. Die schwache Modellierung ist hier als erster Schritt für die menschliche Interpretation ausreichend. Eine gute Definition bietet dem Endnutzer eine Anleitung für die spätere Klassifizierung und Instanziierung seiner Daten entsprechend diesen Begriffen. Diese Ebene dient hauptsächlich dazu, ein einheitliches menschliches Verständnis zu fördern und mindestens sicherzustellen, dass die Daten für Maschinen mit dem geringsten Fähigkeitsniveau oder LoC-1 *verarbeitbar* sind. Begriffe und Definitionen können verbreitet und für andere zur Verweisung und Wiederverwendung auf Webseiten veröffentlicht werden. Für dieses LoC-1 darf nur die RDF- und SKOS-Sprache verwendet werden.

Wo immer Einheitlichkeit für den *Austausch oder das Teilen* von Asset-Daten zwischen digitalen Systemen verschiedener Parteien erforderlich ist, wird eine expressivere Leistung in Form von LoC-2 benötigt, wo Daten nach Ontologien, die Konzepte, Datentypen, Attribute und Relationen oder Beschränkungen enthalten, klassifiziert sind. Diese stärkere Ebene baut auf der untersten Ebene auf und macht die Daten auch für Maschinen *interpretierbar*. Durch die Hinzufügung dieser Semantik wird eine automatische (Open World) Ableitung von Daten aus angegebenen Daten möglich. Für dieses LoC-2 müssen die RDFS- oder (RDFS + OWL)-Modellersprachen verwendet werden.

ANMERKUNG Die OWL wird zusätzlich zu der RDFS definiert und liefert eine „logischere“ Modellierungsleistungsfähigkeit als RDFS durch die Einführung von „Beschränkungen“, die verwendet werden können, um neue Daten aus angegebenen Daten abzuleiten.

Um letztendlich wirklich Daten für alle Arten von Entscheidungsunterstützung *integrieren* zu können, wird eine noch stärkere Semantik benötigt. Diese stärkere Kenntnis liegt in Form ausdrücklicher Beschränkungen und Regeln für die Daten vor, anhand der die Daten in einer geschlossenen Umgebung verifiziert und abgeleitet werden können. Diese Beschränkungen und Regeln können definitorisch oder operativ sein.

Letztere werden durch die Aufgabenstellung des Auftraggebers oder gesetzgebende/regulatorische Organe zusätzlich zu den definitorischen Beschränkungen und Regeln festgelegt. Für dieses LoC-3 muss die (RDFS+) SHACL („Core“ für Überprüfung, „erweiterte Merkmale“ für die Ableitung) Sprache verwendet werden. Für weitere Informationen zu den erweiterten SHACL-Merkmalen ist „let“ noch keine Empfehlung (und daher nicht in den normativen Verweisungen enthalten), siehe [5].

6 Konzeptionelle L1: Datensprache

Die von einer „konkreten Sprache“ unabhängige konzeptionelle L1: Datensprache wie in Tabelle 2 festgelegt.

Tabelle 2 — Konzeptionelle L1: Datensprache

Metasatz		
1. Datenmodell		
2. Datensatz		
3. Gruppe		
Metakonzept		
4. Konzept		
5. Individuum		
6. Wertetyp		
7. Wert		
Metarelation	Quelle	Ziel
8. Attribut		
1. Anmerkung	1 bis 11 (oder)	7
2. Qualität	4 oder 5	7
3. Größe	4 oder 5	7
9. Relation		
1. Gruppierung	3 bis 11 (oder)	1 bis 3 (oder)
	5	1 bis 3 (oder)
2. Klassifizierung (invers: Instanziierung)	5	4
	7	6
3. Generalisierung (invers: Spezialisierung)	4	4
	6	6
	8	8
	9	9
4. Zusammensetzung (invers: Zergliederung)	4	4
	5	5
5. Assoziation	4	4
	5	5
10. Beschränkung		
	4 bis 6 (oder)	8 oder 9
	8 oder 9	4 bis 6 (oder)
11. Ableitung		
	4 bis 6 (oder)	8 oder 9
	8 oder 9	4 bis 6 (oder)
12. Metarolle	4 oder 5	9

Diese konzeptionelle Datensprache enthält Metasätze, Metakonzepte, Metarelationen und eine Metarolle.

Metasätze

Der erste Metasatz repräsentiert eine Spezifikation für Daten, die allgemein als Datenmodell bezeichnet wird. Gute Beispiele für Datenmodelle sind Vokabularien, Thesauri, Wörterbücher und Ontologien. Betrachtungsmodelle, die nur bestehende andere Modelle (ohne Veränderung) wiederverwenden, sind ebenfalls ein Beispiel für Datenmodelle. Darüber hinaus gibt es Datensätze, die individuelle Daten enthalten, die üblicherweise einem Datenmodell entsprechen. Darüber hinaus können benutzerdefinierte Gruppen bestehen, die Teilsätze der Datensätze oder des Datenmodells repräsentieren.

Metakonzepte

Das erste einfache Metakonzept ist ein Element eines Datenmodells und ein Konzept, das auf abstrakte Vorstellungen als Dinge von Interesse verweist. Das nächste Konzept ist ein Individuum, eine Instanz eines Konzepts, die etwas repräsentiert, auf das man in der Realität zeigen kann oder könnte. Das nächste Metakonzept ist ein Datentyp wie ein String, ein boolescher oder ein Aufzählungstyp, der letztendliche Werte als Instanzen hat.

Metarelationen

Es wird zwischen zwei Hauptformen von Metarelationen unterschieden:

- Attribute;
- Relationen.

Attribute beschreiben intrinsische Eigenschaften und Relationen beschreiben extrinsische Eigenschaften einzelner Konzepte. Attribute werden weiterhin in *Anmerkungen* unterteilt, die von Menschen interpretierbare ID, Namen, Label, Definitionen usw. sowie durch Computer interpretierbare *Qualitäten* und *Größen* hinzufügen.

Drei Mechanismen sind als spezifische Relationen definiert:

- Klassifizierung (invers: Instanziierung) von „konkret“ zu „abstrakt“;

Konzepte müssen mit Individuen instanziiert werden, die auf reale Dinge verweisen, auf die man zeigen kann (oder im Prinzip im Fall eines imaginären Objekts wie einem Plan oder einem Design zeigen „könnte“). Solche Instanzen erhalten lexikalische Werte für Attribute oder Verweisungen für Relationen. Lexikalische Werte müssen entsprechend einem Wertetyp (wie etwa String, Dezimalzahl, ganze Zahl, boolesch usw.) klassifiziert werden.

- Generalisierung (invers: Spezialisierung) von „spezifisch“ zu „generisch“;

Konzepte müssen zu anderen Konzepten spezialisiert werden; Attribute müssen zu anderen Attributen spezialisiert werden und Relationen müssen zu anderen Relationen spezialisiert werden. Spezialisierte Konzepte, Attribute und Relationen übernehmen alle Beschränkungen und Ableitungen der Konzepte, Attribute und Relationen, aus denen sie spezialisiert wurden.

- Zusammensetzung (invers: Zergliederung) von „detailliert“ zu „global“;

Konzepte müssen in Konzepte und Instanzen in andere Instanzen zergliedert werden.

Diese drei Mechanismen generieren die drei folgenden Hierarchietypen:

- eine *Typologie* (von Konzepten);
- eine *Taxonomie* (von Konzepten, Attributen oder Relationen);

- eine *Meronomie* (von Konzepten).

Die folgenden weiteren Relationstypen müssen neben den drei Abstraktionsmechanismen angewendet werden:

- Gruppierung wie etwa für die Gruppierung aller Konzepte, Attribute und Relationen usw. in einem Datenmodell;
- allgemeine Assoziationen auf Konzept- und individueller Ebene;
- Beschränkungen, die den Betrag von Werten, die Werte selbst oder beides einschränken. Konzepte, Wertetypen, Attribute und Relationen können Beschränkungen im Hinblick auf ihre Quell- oder Zielkonzepte (im Fall von Relationen) oder den Zielwerttyp (im Fall von Attribut) haben.

Ableitungen, die Auskunft darüber geben, wie neue Werte für Attribute oder Relationen aus bestehenden angegebenen Werten abgeleitet werden müssen.

Beschränkungen müssen „definitivisch“ sein oder Anforderungen, Vorschriften oder Empfehlungen repräsentieren.

Assoziationen in diesem Dokument müssen grundsätzlich n-ische Assoziationen sein. Alle spezifischen Sprachbindungen in diesem Dokument verwenden jedoch nur binäre Assoziationen. Falls n-ische Relationen modelliert werden müssen (genauer gesagt Assoziationen), muss der in [6] beschriebene Ansatz verwendet werden.

Metarolle

Eine Metarolle wird nur im Fall von n-ischen Relationen und wenn kein Verfahren zu ihrer Umwandlung in binäre Relationen (wie in [6]) verwendet wird benötigt. Eine solche Metarolle definiert die Einbindung eines Konzepts oder Individuums in eine Beziehung. Im Fall einer binären/gerichteten Relation wie etwa in einem vernetzten Datenansatz werden solche Rollen nicht benötigt. Die Rolle kongruiert mit der Relation selbst. Dieser Metaaspekt könnte jedoch für potentielle zukünftige Nicht-LD-Sprachbindungen wie zu UML oder dem relationalen Modell hilfreich sein.

7 Konkrete L1: Datensprachbindungen

7.1 Einleitung

Eine Sprachbindung definiert, wie die konzeptionelle Datensprache den verfügbaren Modellierungskonstrukten, die von einer gegebenen konkreten Datensprache bereitgestellt werden, zugeordnet wird. Aus Gründen der Vereinfachung verwendet dieses Dokument wo immer möglich die einfache und direkte Nutzung verfügbarer Sprachkonstrukte. Dies bedeutet, dass nur Größen „objektifiziert“ und als Klassen modelliert werden dürfen, während Qualitäten und Relationen direkt als RDF-Eigenschaften modelliert werden müssen (in OWL: Datentypeeigenschaften und Objekteigenschaften). Auf diese Weise müssen angebotene Standard-Sprachfunktionalitäten wie inverse Relationen und verschiedene Attribut- oder Relationsbeschränkungsattribute oftmals direkt wiederverwendet werden.

Die folgenden Tabellen 3 und 4 enthalten die Spezifikationen für jeden Modellierungsstil je Fähigkeitsniveau (LoC).

Tabelle 3 — Spezifizierte Sprachbindung für LoC-1: Abstimmung der Begriffe und Definitionen mit SKOS

Metasatz/Konzept/Relation	
1. Datenmodell	skos:ConceptScheme
2. Datensatz	nicht anwendbar
3. Gruppe	skos:Collection
4. Konzept	skos:Concept
5. Individuum	im Allgemeinen nicht anwendbar, nur Aufzählungselemente: skos:Concept
6. Wertetyp	nicht anwendbar
7. Wert	nicht anwendbar
8. Attribut	
Anmerkung	bestehende skos-Anmerkungen
Qualität	skos:Concept
Größe	skos:Concept
9. Relation	
Gruppierung	skos:inScheme, skos:member
Klassifizierung/ Instanziierung	im Allgemeinen nicht anwendbar, nur für Aufzählungselemente: skos:broader / skos:narrower
Generalisierung/ Spezialisierung	skos:broader / skos:narrower
Zusammensetzung/ Zergliederung	skos:broader / skos:narrower
Assoziation	skos:Concept
10. Beschränkung	skos:related
11. Ableitung	skos:related
12. Metarolle	n.a.

ANMERKUNG 1 Individuen werden nicht modelliert. Aus diesem Grund sind rdf:type-Relationen nicht relevant. Es gibt eine Ausnahme: die für nummerierte Datentypen verwendeten Instanzen von „type classes“. Hier werden die zulässigen Werte ebenfalls zu SKOS-Konzepten, die mit der allgemeineren Aufzählungsklasse verbunden sind.

ANMERKUNG 2 Technische Darstellungseinheiten, die eine räumliche und zeitliche Platzierung beinhalten, sind ebenfalls nicht relevant.

Tabelle 4 — Spezifizierte Sprachbindungen für LoC-2&3: Datenaustausch/-teilung oder -integration

Metasatz/ Konzept/ Relation	Sprachbindung für LoC-2a: Datenaustausch/-teilung mit Open-World-RDFS	Sprachbindung für LoC-2b: Datenaustausch/-teilung mit Open World (RDFS+) OWL	Sprachbindung für LoC-3: Datenintegration in Closed World (RDFS+) SHACL
1. Datenmodell	owl:Ontology		
2. Datensatz	owl:Ontology		
3. Gruppe	rdfs:Container		
4. Konzept	rdfs:Class	owl:Class	rdfs:Class
5. Individuum	rdfs:Resource (implizit)	owl:NamedIndividual oder anonymes Individuum (implizit)	rdfs:Resource (implizit)
6. Wertetyp	rdfs:Datatype oder rdfs:Class + rdf:Property für Aufzählungs- Datentypen	rdfs:Datatype oder owl:Class + owl:ObjectProperty für Aufzählungs-Datentypen; einschließlich owl:oneOf für feste Listen	rdfs:Datatype oder rdfs:Class + rdf:Property für Aufzählungs- Datentypen
7. Wert	Klartext oder Buchstabensymbol (implizit)	Klartext oder Buchstabensymbol	Klartext oder Buchstabensymbol (implizit)
8. Attribut			
Anmerkung	rdf:Property	owl:AnnotationProperty	rdf:Property
Qualität	rdf:Property (außer für Aufzählung) enumeration: rdf:Property mit Bereich einer Teilklassse von smls- rdfs: EnumerationType-Klasse	owl:DatatypeProperty (außer für Aufzählungen) Aufzählung: owl:ObjectProperty mit Bereich einer Teilklassse von smls-owl: EnumerationType-Klasse	rdf:Property (außer für Aufzählungen) Aufzählung: rdf:Property mit sh:PropertyShape mit sh:class Teilklassse von smls-shacl: EnumerationType
Größe	rdf:Property mit Bereich smls-rdfs: QuantityValue	owl:ObjectProperty mit Bereich smls-owl: QuantityValue	rdf:Property mit sh:PropertyShape mit sh:class smls- shacl:QuantityValue
9. Relation			
Gruppierung	implizit (gleiche Datei) oder rdfs:member		
Klassifizierung/ Instanziierung	rdf:type		
Generalisierung/ Spezialisierung	rdfs:subClassOf oder rdfs:subPropertyOf	rdfs:subClassOf & rdfs:subPropertyOf	rdfs:subClassOf & rdfs:subPropertyOf

Metasatz/ Konzept/ Relation	Sprachbindung für LoC-2a: Datenaustausch/-teilung mit Open-World-RDFS	Sprachbindung für LoC-2b: Datenaustausch/-teilung mit Open World (RDFS+) OWL	Sprachbindung für LoC-3: Datenintegration in Closed World (RDFS+) SHACL
Zusammen- setzung/ Zergliederung	nicht verfügbar: zur Modellierung über smls- rdfs:hasPart (Instanzebene, beschränkt auf Klassenebene)	nicht verfügbar: zur Modellierung über smls- owl:hasPart (Instanzebene, beschränkt auf Klassenebene)	nicht verfügbar: zur Modellierung über smls-shacl:hasPart (Instanzebene, beschränkt auf Klassenebene)
Assoziation	rdf:Property	owl:ObjectProperty	rdf:Property
10. Beschränkung	rdfs:domain oder rdfs:range	owl:Restriction (+ Details) rdfs:domain & rdfs:range	sh:NodeShape & sh:PropertyShape ⁴
11. Ableitung	nicht anwendbar	nicht anwendbar	sh:Rule
12. Metarolle	n.a.	n.a.	n.a.

ANMERKUNG 3 Da jede genormte computer-interpretierbare Modellierung tatsächlich mit LoC-2a beginnt, dient diese oftmals als Ausgangspunkt, der in einer späteren Phase auf LoC-3 für eine erweiterte Beschränkungs-/Ableitungsmodellierung erweitert werden könnte.

ANMERKUNG 4 In diesem Dokument wird Turtle für die Codierung von SML selbst verwendet (Anhang A). Beispiele sind für RDF-XML, Turtle und JSON-LD angegeben (Anhang C bis Anhang F).

Die folgenden Modellierungsmuster müssen für die Zuordnung zu einer bestimmten Sprache angewendet werden.

7.2 Identifizierung: URI-Strategie

Die URI-Strategie muss wie in der folgenden Tabelle 5 angegeben festgelegt werden.

4 Wenn eine Domäne *und* ein Bereich besteht: Zuordnung zu einer NodeShape mit einer PropertyShape mit einer sh:targetClass. Wenn nur ein Bereich besteht: Zuordnung zu einer NodeShape mit sh:targetObjectsFrom.

Tabelle 5 — Der festgelegte URI hat die folgende allgemeine Form

uri := https://{internet domain}/{path}/{reference}, dabei ist reference := (def ^a /{data model name}[(/ #){concept}]{attribute}{relation}{reference individual}) (dat ^b /{data set name}[(/ #){individual}]) (doc ^c /{document name}[document extension][(/ #){identifier}])	
Metasymbole	
xyz	Ein vorgegebener String
:=	definiert durch („Produktionsregel“)
{ ... }	obligatorische Komponente (auszufüllen)
[...]	optionale Komponente (auszufüllen)
(...)	Gruppierung
	Logik exklusiv ODER Operator
^a Zur Bezeichnung eines Datenmodells, Konzepts, Attributs, einer Relation oder eines Referenz-Individuums. ^b Zur Bezeichnung eines Datensatzes oder eines realen Individuums. ^c Zur Bezeichnung eines Dokuments mit Informationen über ein Datenmodell, einen Datensatz, ein Attribut, eine Relation oder ein (Referenz-)Individuum.	

BEISPIEL 1 <http://www.interlink.eu/data/def/roadotl#Road>

BEISPIEL 2 <http://www.roadnetwork.nl/id/nwb#A16>

BEISPIEL 3 <http://www.interlink.eu/doc/RoadDefinition.docx>

ANMERKUNG 1 „Datenmodell“ und „Datensatz“ sind optionale Fragmente, da sie manchmal durch die Domäne und/oder den Pfad abgedeckt sind.

ANMERKUNG 2 Die Entscheidung zwischen „/“ oder „#“ hängt von der erwarteten Größe der Datenmodelle/Datensätze ab. Bei großen Modellen wird „/“ verwendet, weitere Informationen können [4] entnommen werden.

ANMERKUNG 3 Falls eine Versionierung in dem URI erforderlich ist, wird der folgende Ansatz festgelegt (versionslose und versionierte Varianten, wobei sich die versionslose Variante immer auf die aktuellste Version bezieht):

— uri := http [s]://{internet domain}/{path}/{version}/{x}

ANMERKUNG 4 Die URI dienen vorzugsweise nicht nur der Kennzeichnung, sondern auch der Lokalisierung (um Datenmodelle, Datensätze und ihre Elemente „dereferenzierbar“ zu machen).

ANMERKUNG 5 Dateierweiterungen dürfen nicht in Namen von Datenmodellen und Datensätzen verwendet werden. Eine standardmäßige „Server Negotiation“ muss für die Auswahl der korrekten Darstellung verwendet werden (einschließlich für Menschen lesbarer Einleitungen, d. h. .html).

ANMERKUNG 6 Falls erforderlich, müssen Internetdienste, die stabile URI bieten, wie w3id.org oder purl.org, verwendet werden.

7.3 Namenskonventionen

Es gibt zwei Hauptmethoden, Dinge in Datensätzen oder Datenmodellen eindeutig zu benennen/kennzeichnen:

- durch sinnvolle Namen oder
- durch semantikfreie Opaque Codes (wie etwa generierter UUID-Code).

Beide Ansätze haben ihre Vor- und Nachteile und sind durch dieses Dokument zugelassen. Für Menschen lesbare Namen helfen dabei, Ontologien zu entwickeln und zu lernen, während in der Betriebsphase mit Softwaretools, die Details verbergen, Codes stabiler (sie bleiben stabil, wenn sich die Begriffe ändern) und sprachneutral sind. Codes ermöglichen es darüber hinaus einen Label-Begriff zu haben, der auf verschiedene Konzepte verweist.

Falls Codes verwendet werden, müssen ebenfalls für Menschen lesbare RDFS- oder SKOS-Label verwendet werden.

Im Fall von Codes müssen mindestens eindeutige bevorzugte Label im Kontext einer Ontologie verwendet werden, da alternative Label nicht eindeutig sind.

Wenn Namen in Datenmodellen verwendet werden:

- Konzept- oder Wertetyppnamen beginnen immer mit einem Großbuchstaben („CamelCase“);
- Attribut- oder Relationsnamen beginnen immer mit einem Kleinbuchstaben („camelCase“);
- Namen sind vorzugsweise in englischer Sprache und im Singular.

Beispielnamen sind in Datensätzen flexibel, z. B.:

- spezifische Namen wie etwa „A16“ für eine niederländische Straße;
- nummerierte Individuen für eine Klasse wie „Road_16“;
- semantikfreie zugewiesene GUID wie „233hhjj334--!!99“; oder
- semantikfreie generierte UUID wie „550e8400-e29b-41d4-a716-446655440000“ (üblicherweise ein Bit lang, jedoch mit dem Vorteil der automatischen Generierung); oder
- jede Kombination der obigen Optionen.

7.4 Anmerkung

Textliche Definitionen müssen das Simple Knowledge Organization System (SKOS) wiederverwenden

- <http://www.w3.org/2004/02/skos/core#definition> (rdf:type rdf:Property)

Dies ermöglicht eine etwas spezifischere Bedeutung als der allgemeinere rdfs:comment.

Und durch die folgende Spezifikation:

rdfs:isDefinedBy muss verwendet werden, um auf eine externe Ressource zu verweisen, die die Subject Resource definiert. Diese Eigenschaft darf angewendet werden, um ein RDF-Vokabular anzuzeigen, in dem eine Subject Resource definiert ist. Sie weist auf maßgebliche Informationen über die Ressource hin (die nicht notwendigerweise RDF ist, sondern oftmals html und in einigen Fällen native Formate wie PDF hat).

Codes (unabhängig von irgendeiner Sprache) müssen wieder mit dem SKOS definiert werden:

— <http://www.w3.org/2004/02/skos/core#notation> (rdf:type owl:DatatypeProperty)

Jede(r) der verwendeten Codes/Notationen muss die Dinge, die den Code/die Notation besitzen, kennzeichnen.

Tabelle 6 enthält eine Übersicht der durch dieses Dokument festgelegten ausgewählten Anmerkungstypen.

Tabelle 6 — Festgelegte Sprachbindung für Anmerkungen

Anmerkung	Sprachbindung
Primäres Label für	skos:prefLabel (deterministisch)
Synonym für	skos:altLabel
Code für	skos:notation
Anmerkung des Autors	skos:editorialNote
Anwendungsbereich	skos:scopeNote
Kommentare	rdfs:comment
Externe Informationen	rdfs:seeAlso
Beispielanwendung	skos:example
Intern definiert durch	skos:definition
Extern definiert durch	rdfs:isDefinedBy
Abgekürzt durch	smls:abbreviation (eine Teileigenschaft von skos:altLabel)

Für einen gemeinsamen Eindruck in Betrachtern/Bearbeitungsprogrammen legt dieses Dokument fest, dass Labels:

- im Fall von Konzepten und Datentypen mit einem Großbuchstaben (Großschreibung) beginnen;
- im Fall von Attributen und Relationen mit einem Kleinbuchstaben (Kleinschreibung) beginnen;
- keine anderen Großbuchstaben verwenden (außer bei Akronymen);
- Leerzeichen zur Worttrennung verwenden;
- keine Punkte am Ende verwenden;
- immer Sprachtags (möglicherweise einschließlich Regionen) nach [3] verwenden. Beispiele: „Bridge“@en, „Pechstrook“@nl-BE;
- Im Fall benutzerfreundlicher Namen werden diese auch als bevorzugtes Label unter Anwendung der obigen Regeln kopiert (wenn der Klassenname „InnerDoor“ lautet, wird für das bevorzugte SKOS-Label ebenfalls „Inner door“ verwendet).

7.5 Aufzählungsdatentypen

Aufzählungsdatentypen sind als Konzept festgelegt, sodass zulässige Aufzählungselemente zu Individuen mit mehrsprachigen Labels und anderen Metadaten wie Beschreibungen, Definitionen usw. werden können. Es wird zwischen offenen und festen Aufzählungsdatentypen unterschieden. Feste Aufzählungsdatentypen werden mit dem Open-World-Konstrukt „owl:oneOf“ und einem Close-World-Gegenstück „sh:in“ festgelegt.

Der Name des Aufzählungsdatentyps ist eine mit einem Großbuchstaben beginnende Variante der Beziehung, die diesen Datentyp als Bereich hat. Aus Gründen der Klarheit wird eine Erweiterung mit einem „Type“-Fragment empfohlen.

BEISPIEL

```
ex:loadLevel
  a owl:ObjectProperty ;
  rdfs:domain ex:Vehicle ;
  rdfs:range ex:LoadLevelType ;
.
ex:LoadLevelType
  a owl:Class ;
  rdfs:subClassOf smls-owl:EnumerationType ;
  owl:oneOf (
    ex:Light
    ex:Normal
    ex:Heavy
  ) ;
.
```

ANMERKUNG Aufzählungselemente können mehreren Aufzählungsdatentypen angehören.

7.6 Zergliederung (Instanz-Ebene)

Dieses Dokument legt eine nicht-transitive hasPart-Relation fest, die nicht notwendigerweise invers funktional ist und eine implizite inverse Beziehung hat. Die transitive Variante muss gegebenenfalls abgeleitet werden. Eine Meronomie (übliche Zergliederung) muss durch Beschränkung dieser hasPart-Relation definiert werden.

ANMERKUNG Dieser Abstraktionsmechanismus ist dahingehend speziell, dass er nicht direkt durch bereits verwendete Sprachen, wie aus früheren Sprachbindungen hervorgeht, unterstützt wird.

7.7 Arten und Einheiten von Größen

Dieses Dokument legt die von qudt.org für Arten und Einheiten von Größen veröffentlichte QUDT [2] Ontologie fest.

ANMERKUNG QUDT ist vollständig mit der Norm ISO/IEC 80000 abgestimmt (System, Namen, Definitionen, Symbole usw.).

7.8 Größenmodellierung

Es bestehen zahlreiche Möglichkeiten, Attribute zu modellieren (Qualitäten oder Größen). Jede dieser Methoden hat jetzt oder in Zukunft Vor- und Nachteile. Die einfachste und direkteste Modellierung eines Attributs z. B. in OWL erfolgt durch ein owl:DatatypeProperty. Bei dieser Methode ist es jedoch schwierig Metadaten zu modellieren, wie etwa die oftmals gewünschte Einheit für einen Wert im Fall einer Größe.

Aus diesem Grund legt dieses Dokument Größen als Beziehungen (wie in OWL als owl:ObjectProperty) mit einem Bereich (rdfs:range) einer vorgegebenen smls:QuantityValue-Klasse fest.

Dieser Größenwert muss einen Wert durch rdf:value erhalten (mit dem Bereich xsd:decimal, aufgezeichnet durch eine Beschränkungen der Größenwertklasse), möglicherweise eine Einheit durch smls:unit eines anderen Meta-Attributs. Metagrößen selbst werden auf die gleiche Weise behandelt (d.h. eine verschachtelte/rekursive Anwendung). Bild 3 zeigt ein übliches Beispiel.

ANMERKUNG 1 Qualitäten werden genau wie bei üblichen Beziehungen einfach gehalten; in der Praxis besteht weniger Bedarf an Metadaten über Qualitäten/Beziehungen (sicherlich keine Einheiten).

ANMERKUNG 2 Zusätzlich zu den Modellierungs-Metadaten über die Instanziierung/den Wert der Größe werden auch oftmals Metadaten über die Größendefinition selbst benötigt. Hierzu werden Eigenschaften für Eigenschaften benötigt (dies führt zu OWL Full).

ANMERKUNG 3 Da Größenwerte nicht wiederverwendet werden müssen, können sie auf der Instanzebene anonym bleiben (ein sogenannter „Blindwert-Knoten“). Darüber hinaus muss die Instanz nicht als ein smls:QuantityValue eingegeben werden. Diese Eingabe kann abgeleitet werden, indem jedem Attribut der Bereich smls:QuantityValue zugewiesen wird.

BEISPIEL (graphisch)

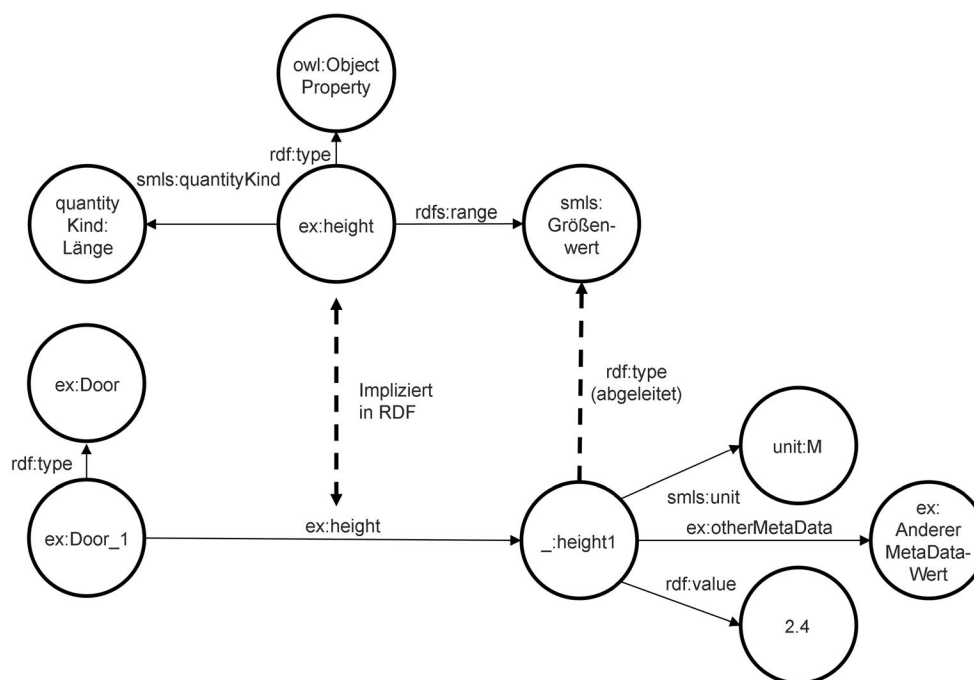


Bild 3 — Beispiel-Größenmodellierung

7.9 Gruppierung

Ein Gruppierungsmechanismus wird durch rdfs:Container und rdfs:member hinzugefügt. Dieser Standardmechanismus muss angewendet werden, um beispielsweise Attributdefinitionen wie etwa die Größendefinitionen „Höhe“, „Breite“ und „Tiefe“ in „geometrischen Größen“ zu gruppieren. Auf diese Weise kann auch die Gruppierung auf Instanzebene abgeleitet werden.

8 Generisches M1: Datenmodell der obersten Ebene

8.1 Modell der obersten Ebene

Dieses Dokument legt die folgenden Konzepte der obersten Ebene und ihre Wechselbeziehungen (eigentlich: Assoziationen) und (hasPart) Beschränkungen fest:

- Einheit
 - Objekt
 - PhysicalObject
 - InformationObject
 - Aktivität
- Status
- Ereignis
- SpatialRegion
- TemporalRegion

Bei Objekten (unterteilt in physische Objekte und Informationsobjekte) wird zwischen *existierenden* Objekten und sich *ereignenden* Aktivitäten unterschieden. Das zuvor vorgestellte Meta-Modell führt noch keine „Zeitaspekte“ ein, sodass dies durch die eigentliche konzeptionelle Modellierung erfolgen muss; hier durch die Einführung dynamischer Konzepte über die statischen Objekte und Aktivitäten als Zustände und Ereignisse hinaus. Wenn Zeit im Fall zeitloser statischer Aspekte nicht relevant oder nur ein „Schnappschuss“ von Objekten in der Zeit ist, können diese Zustände und Ereignisse ignoriert (impliziert gehalten) werden.

Es bestehen zwei Hauptbeziehungen zwischen Objekten und Aktivitäten: Physische Objekte führen Aktivitäten aus und Aktivitäten werden durch physische Objekte umgewandelt.

Einheiten – d.h. Objekte und Aktivitäten – existieren in Raum und Zeit und müssen einen relevanten Innenraum und eine Grenze in Raum und Zeit haben (über die räumliche und zeitliche Region).

Eine räumliche Region muss für zwei Zwecke verwendet werden: die Definition des topologischen Innenraums und der Grenze eines physischen Objekts. So wird beispielsweise durch die Festlegung der Grenze eines funktionalen Transportnetzwerks ein 3D-Korridornetzwerk definiert. Dies muss erreicht werden, indem die Grenzen der Kanten innerhalb des Netzwerks, die selbst topologisch durch Knoten verbunden sind, aggregiert werden.

Dieses Dokument legt die folgenden grundlegenden Modellierungskonstrukte zur Definition einer Position in Raum und Zeit fest:

- hasSpatialLocation;
- hasSpatialReferenceSystem;
- hasSpatialReference;
- SpatialReferenceSystem;

- SpatialReference;
- hasTemporalLocation;
- hasTemporalReferenceSystem;
- hasTemporalReference;
- TemporalReferenceSystem;
- TemporalReference.

Darüber hinaus müssen einige spezifische Formen räumlicher und zeitlicher Bezugssysteme vordefiniert werden.

Des Weiteren muss das Modell der obersten Ebene Relationen „derselben Ebene“ (Assoziationen) haben, die auf ihre Quell- und Zielkonzepte einschließlich Klassenbeschränkungsvarianten des Zusammensetzungs-/Zergliederungsmechanismus beschränkt sind (auch ihre Umkehrungen sind gegeben). Diese sind in Tabelle 7 und ihre Umkehrungen in Tabelle 8 angegeben.

Tabelle 7 — Festgelegte Relationen der obersten Ebene

Assoziationen		
[InformationObject]	beschreibt	[All]
[PhysicalObject]	hasState	[State]
[PhysicalObject]	löst aus	[Event]
[PhysicalObject]	führt aus	[Activity]
[PhysicalObject]	hasPeriod	[TemporalRegion]
[PhysicalObject]	hasInterior	[SpatialRegion]
[PhysicalObject]	hasBoundary	[SpatialRegion]
[Activity]	hasState	[State]
[Activity]	wandelt um	[PhysicalObject]
[Activity]	wandelt um	[InformationObject]
[Activity]	löst aus	[Event]
[Activity]	hasPeriod	[TemporalRegion]
[Activity]	hasInterior	[SpatialRegion]
[Activity]	hasBoundary	[SpatialRegion]
[Event]	startet	[State]
[Event]	beendet	[State]
[State]	hasPeriod	[TemporalRegion]
[State]	hasInterior	[SpatialRegion]
[State]	hasBoundary	[SpatialRegion]
Zusammensetzung/Zergliederung		
[PhysicalObject]	hasPart	[PhysicalObject]
[InformationObject]	hasPart	[InformationObject]
[Activity]	hasPart	[Activity]

Tabelle 8 — Festgelegte inverse Relationen der obersten Ebene

Inverse Assoziationen		
[All]	isDescribedBy	[InformationObject]
[State]	isStateOf	[PhysicalObject]
[Event]	isTriggeredBy	[PhysicalObject]
[Activity]	isPerformedBy	[PhysicalObject]
[TemporalRegion]	isPeriodFor	[PhysicalObject]
[SpatialRegion]	isInteriorOf	[PhysicalObject]
[SpatialRegion]	isBoundaryOf	[PhysicalObject]
[State]	isStateOf	[Activity]
[PhysicalObject]	isTransformedBy	[Activity]
[InformationObject]	isTransformedBy	[Activity]
[Event]	isTriggeredBy	[Activity]
[TemporalRegion]	isPeriodFor	[Activity]
[SpatialRegion]	isInteriorOf	[Activity]
[SpatialRegion]	isBoundaryOf	[Activity]
[State]	isBegunBy	[Event]
[State]	isEndedBy	[Event]
[TemporalRegion]	isPeriodFor	[State]
[SpatialRegion]	isInteriorOf	[State]
[SpatialRegion]	isBoundaryOf	[State]
Zusammensetzung/Zergliederung		
[PhysicalObject]	isPartOf	[PhysicalObject]
[InformationObject]	isPartOf	[InformationObject]
[Activity]	isPartOf	[Activity]

Durch graphische Kombination werden die Konzepte und Assoziationen des Modells der obersten Ebene (Blatt) in Bild 4 erhalten.

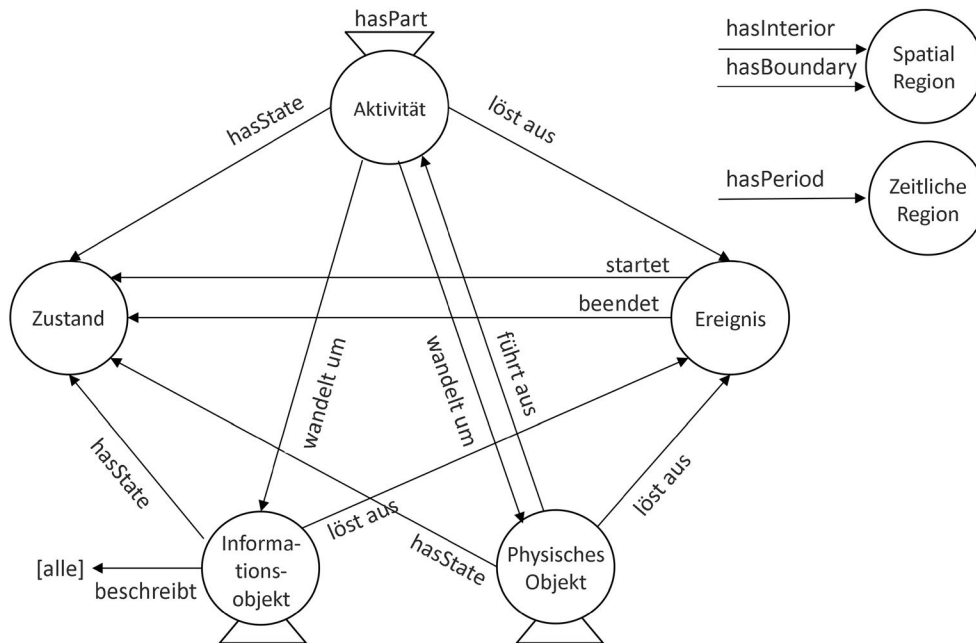


Bild 4 — Kombinierte vordefinierte Konzepte und Relationen

8.2 Systems-Engineering-Erweiterung

8.2.1 Einleitung

Alle zuvor unterschiedenen archetypischen Konzepte (physisches Objekt, Informationsobjekt, Aktivität, Zustand und Ereignis) können weiter entlang zweier Dimensionen unterklassifiziert werden. Physische Objekte müssen entsprechend diesem Dokument weiter unterklassifiziert werden:

- funktionales Objekt oder technisches Objekt;
- geplantes Objekt oder gebautes Objekt.

ANMERKUNG 1 Wenn ein Individuum als eine Art klassifiziert wird, gilt dies üblicherweise für seine gesamte Lebensdauer. Wenn ein Objekt zu einem Zeitpunkt ein Buch ist, bleibt es üblicherweise immer ein Buch. Es geht darum, was etwas „ist“. Wenn ein Individuum als eine Rolle klassifiziert wird, ist es oftmals mit einem bestimmten Zeitraum verbunden. Wenn jemand ein Vater ist, ist er das üblicherweise ab einem bestimmten Anfangszeitpunkt. Jemand „ist“ also kein Vater, sondern spielt die Rolle eines Vaters. Er könnte diese Rolle auch mehrfach gleichzeitig oder nacheinander spielen.

ANMERKUNG 2 Diese Dimension ist vollständig orthogonal mit „üblicher“ Zergliederung (auch bekannt als Zergliederung der Klassenebene): Die Unterscheidung kann auf jeder Skalenebene vorgenommen werden: Netzwerke, Komplexe, Assets, Element, Komponenten und Materialien.

ANMERKUNG 3 Funktionale Objekte können durch einen Auftraggeber in teilfunktionale Objekte zergliedert werden, wobei die Blätter des Zergliederungsbaums von Auftragnehmern/Lieferanten, die hergestellte/beschaffte Lösungen anbieten, die wiederum selbst in teiltechnische Objekte zergliedert sind, implementiert werden. Dieser Prozess kann selbstverständlich rekursiv sein und mehrere „FO-TO“-Sprünge durch anschließende Parteien in der Lieferkette beinhalten.

Wenn die beiden Dimensionen kombiniert werden (Bild 5), werden impliziert vier weitere Teilklassen physischer Objekte „generiert“. Die Pfeile in den Quadranten zeigen den üblichen Prozessablauf in der Praxis: Ein Asset wird funktional programmiert, technisch gestaltet und technisch so gebaut, dass es in der Realität die vorgesehene Funktion erfüllt. Das Asset kann zu einem späteren Zeitpunkt umprogrammiert, umgestaltet oder umgebaut werden (zum Ende der Lebensdauer schließlich zerstört und verwertet werden).

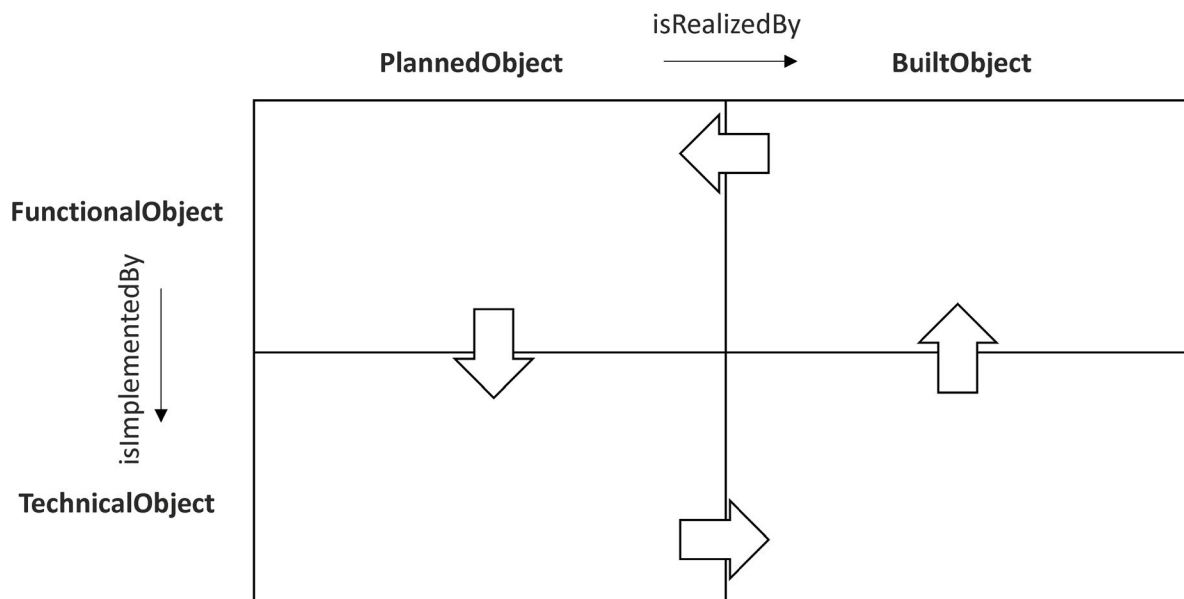


Bild 5 — Vier relevante Quadranten für das Systems Engineering

Ein funktionales Objekt muss eine Funktion als Teilklasse des bestehenden Aktivitäts-Archetyps erfüllen. Wie eine Aktivität nimmt auch diese Funktion immer die Form eines Verbs (kein Substantiv) wie „verbinden“, „pumpen“ oder „kaufen“ usw. an.

Zusammenfassung der SML-Konstrukte:

- smls:FunctionalObject;
- smls:TechnicalObject;
- smls:PlannedObject;
- smls:BuiltObject;
- smls:isImplementedBy (von smls:FunctionalObject zu smls:TechnicalObject);
- smls:isRealizedBy (von smls:PlannedObject zu smls:BuiltObject).

8.2.2 Konformität mit Anforderungen

Funktionale Objekte (imaginär oder real) müssen funktionale Anforderungen haben (unstrukturiert/textlich und/oder strukturiert). Technische Objekte (imaginär oder real) müssen technische Anforderungen haben, die auch als „technische Grenzbedingungen“ (ebenfalls unstrukturiert/textlich und/oder strukturiert) bekannt sind. Funktionale Objekte (imaginär oder real) und technische Objekte (imaginär oder real) können Beobachtungen haben. Beobachtungen für imaginäre Objekte behandeln geplante, gestaltete, berechnete und simulierte Aspekte. Beobachtungen für reale Objekte behandeln reale Aspekte wie Messungen, Prüfungen, Inspektionen usw.

Die Bewertung von Anforderungen anhand von Beobachtungen führt zu Konformitätsniveaus. Die Prüfung zwischen Anforderungen und imaginären Beobachtungen („geplante Leistung“) wird als „Verifizierung“ bezeichnet. Die Prüfung zwischen (verifizierten) imaginären Beobachtungen und realen Beobachtungen („reale Leistung“) wird als „Validierung“ bezeichnet. Eine unzureichende Konformität führt zu einem Entscheidungsfindungsprozess, der Varianten und bevorzugte Lösungen auf der Grundlage einer Multi-Stakeholder-/Multikriterien-Analyse, die zu einer Entscheidung über die Anwendung von Maßnahmen/Aktivitäten führt, d.h. eine technische (Lebenszyklus) Aktivität: Instandhaltung (Art von Umbau),

Renovierung (einschließlich Umgestaltung) usw. oder eine funktionale Aktivität (wie das Überdenken der funktionalen Anforderungen).

Eine technische Aktivität (Lebenszyklus) führt zu einer Veränderung des Zustands eines technischen Objekts, das (um-)gestaltet und/oder (um-)gebaut wird. Eine funktionale Aktivität führt zu einer Veränderung des Zustands eines funktionalen Objekts das (um-)genutzt wird, wobei die letzte Umnutzung „kein Zweck mehr“ oder auch der Abriss (und die potentielle Verwertung) des Objekts ist.

Dieses Dokument legt die folgenden Konstrukte für die Modellierung unstrukturierter Anforderungen fest:

- hasRequirement;
 - Anforderung;
 - hasRequirementSpecification;
 - hasRequirementSource (Auswahl aus: durch Definition, durch den Auftraggeber, per Gesetz oder Verordnung oder durch Empfehlung);
 - hasRequirementsSeverity (Auswahl aus: Bedarf oder Wunsch).

9 Implementierung von SML in Code

Alle obigen Spezifikationen werden für alle drei LoC in den vier Sprachen im normativen Anhang A implementiert. In allen Fällen wird das Turtle-Format verwendet. Die vier festgelegten konkreten Sprachteilsätze werden in dem normativen Anhang B festgelegt.

ANMERKUNG Die Anhänge C bis F enthalten informative Beispiele für alle festgelegten Sprachen und Formate.

10 Vernetzung von Daten

10.1 Arten von Vernetzungen

Die Modellierung von Datenmodellen (hier Ontologien) und Datensätzen ist eine Sache. Die Vernetzung von Datenmodellen, die Vernetzung von Datensätzen und die Vernetzung von Datensätzen mit Datenmodellen ist eine andere Sache. Diese Vernetzung wird in Zukunft an Bedeutung gewinnen, weil Datenmodelle und Datensätze in der Praxis oftmals in einer Wechselbeziehung stehen und sich gegenseitig in der integrierten Entscheidungsfindung ergänzen. Die Idee besteht darin, Netzwerke oder Ökosysteme von Daten und Datenstrukturen zu bilden, die die Grundlage für eine sektorweite Digitalisierung, Integration und Innovation als digitales System bilden.

Diese Vernetzung kann auf verschiedene Arten stattfinden:

- implizit, beispielsweise über eine SPARQL-Abfrage, die mehrere Ontologien und/oder Datensätze abfragt. Die Vernetzungen sind dann implizit in dem SPARQL-Code impliziert.
- explizit (und daher besser wiederverwendbar) durch die deklarative Modellierung der Vernetzungen, vorzugsweise separat zu den in „Vernetzungssätzen“ zu vernetzenden Ontologien/Daten. Auf diese Weise bleiben die Vernetzungssätze/Vernetzungen „agnostisch“ und „sauber“.

Technisch und aus Sicht einer vernetzten Datentechnologie ist ein einfacher Vernetzungssatz eine einfache Sammlung von RDF-Dreiergruppen, in denen das Objekt und Subjekt der Dreiergruppen auf Elemente anderer Ontologien oder Datensätze verweisen. Jede rdf:Property-Instanz kann dann als eine „Vernetzung“ betrachtet werden, sodass alle Beziehungen, die in der „Modellierung“ verwendet werden, ebenfalls für die „Vernetzung“ verwendet werden können.

BEISPIELE

- 1) disk:Bridge rdfs:subClassOf cbnl:Span
- 2) bdb:VanBrienoord owl:sameAs cdb:Brienoord
- 3) ex:BuildingTNO-Delft smls:hasPart bim:Room-E3.64
- 4) rot:MaasTunnel rdf:type rwsotl:Tunnel
- 5) ex1:Asphalt ex2:testedBy kvk:TNO

10.2 Sprachvernetzungsätze auf Sprachebene

Die Möglichkeiten der Vernetzung von Ontologien und Datensätzen hängen zu einem großen Teil von dem Fähigkeitsniveau und der entsprechenden angewendeten vernetzten Datensprache ab. Nachfolgend wird festgelegt, welche Sprachebenen-Konstrukte für jedes Fähigkeitsniveau verwendet werden müssen.

LoC-1 „schwache“ semantische Vernetzung:

- skos:exactMatch;
- skos:closeMatch;
- skos:narrowMatch;
- skos:broadMatch;
- skos:relatedMatch;
- rdfs:seeAlso.

LoC-2/3 „starke“ semantische Vernetzung:

- rdfs:subClassOf zwischen Klassen;
- rdfs:subPropertyOf zwischen Eigenschaften;
- via rdfs:domain oder rdfs:range (d. h. Verweisung auf eine Domäne oder eine Bereichsklasse/einen Datentyp in einer externen Ontologie);
- + die „Kurzschrift“: owl:equivalentClass und owl:equivalentProperty;
- owl:sameAs zwischen Individuen auf Datenebene.

ANMERKUNG Vernetzungen auf Datenebene werden vorzugsweise durch die Klassifizierung eines Individuums mit mehreren rdf:type-Beziehungen zu RDFS/OWL-Klassen aus verschiedenen Ontologien auf der Grundlage des Prinzips der einzelnen Faktenquelle (SSoF) vermieden. Dies ist oftmals nicht möglich, weil mehrere unabhängige Datenquellen von unterschiedlichen Datenquelleninhabern verwaltet werden, die etwas über dasselbe Phänomen in der realen Welt aussagen.

11 Konformität

11.1 Allgemeines

Dieser Abschnitt definiert zwei Konformitätsniveaus für Datensätze im Hinblick auf diese SML-Norm.

11.2 Konformität auf Sprachebene

Ein Datensatz ist auf Sprachebene konform, wenn er wie durch dieses Dokument vorgeschrieben:

- 1) In Übereinstimmung mit einem Standard-RDF-Format (RDF/XML, Turtle oder JSON-LD) ausgedrückt wird und/oder direkt über die Standard-Abfragesprache SPARQL 1.1 zugänglich ist.
- 2) Im Fall expliziter Datenmodelle, die die Datensätze „semantisch“ machen, werden die Standardsprachen für vernetzte Daten (wie RDF, SKOS, RDFS, OWL und/oder SHACL) und die allgemeinen Modellierungs- und Vernetzungsmuster nach dieser Europäischen Norm verwendet. Dies setzt voraus, dass diese Datenmodelle ebenfalls entsprechend den für die Datensätze verwendeten Standard-RD-Formaten ausgedrückt werden.

ANMERKUNGEN

Äquivalente Formate sind beispielsweise RDF/XML, Turtle und JSON-LD.

Datenmodelle geben Assetdaten aus jeder Perspektive wieder, einschließlich:

- softwarespezifische Semantik;
- personenspezifische Semantik;
- projektspezifische Semantik;
- abteilungsspezifische Semantik;
- organisationsspezifische Semantik;
- landesspezifische Semantik;
- vollständig generische internationale Semantik.

Semantik kann weiterhin softwarespezifisch sein, aber diese Semantik wird mindestens explizit modelliert und ist daher leichter zu jeder anderen Semantik umwandelbar. Aus diesem Grund sind Daten auf C2 syntaktisch und semantisch „von der Software befreit“.

11.3 Konformität auf semantischer Ebene

Ein Datensatz ist auf semantischer Ebene konform, wenn er das generische Datenmodell der obersten Ebene (ausgedrückt in einem der Standard-RDF-Formate und unter Verwendung der Standardsprachen für vernetzte Daten) nach dieser Europäischen Norm wiederverwendet.

C2 ist nur zusätzlich zu C1 sinnvoll.

Das tatsächlich in der Praxis erforderliche Konformitätsniveau kann in einer Austausch-Informationsanforderung (EIR) festgelegt werden.

Anhang A (normativ)

Konzeptionelle Datensprache und Datenmodell der obersten Ebene

A.1 Allgemeines

Dieser Anhang beschreibt den Standard-SML-Implementierungscode in den ausgewählten Sprachen für vernetzte Daten (SKOS, RDFS, OWL und SHACL) in der konkreten Syntaxform Turtle. Er behandelt die konzeptionelle Datensprache und das allgemeine Datenmodell der obersten Ebene.

A.2 in SKOS (Turtle-Format)

```
# baseURI: https://w3id.org/def/smls-skos
# imports: http://www.w3.org/2004/02/skos/core
# prefix: smls-skos

@prefix smls-skos: <https://w3id.org/def/smls-skos#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<https://w3id.org/def/smls-skos>
  a skos:ConceptScheme ;
  owl:imports <http://www.w3.org/2004/02/skos/core> ;
  skos:hasTopConcept smls-skos:Entity ;
  skos:hasTopConcept smls-skos:State ;
  skos:hasTopConcept smls-skos:Event ;
  skos:hasTopConcept smls-skos:RequirementSourceType ;
  skos:hasTopConcept smls-skos:RequirementSeverityType ;
  skos:hasTopConcept smls-skos:begins ;
  skos:hasTopConcept smls-skos:ends ;
  skos:hasTopConcept smls-skos:triggers ;
  skos:hasTopConcept smls-skos:transforms ;
  skos:hasTopConcept smls-skos:performs ;
  skos:hasTopConcept smls-skos:describes ;
  skos:hasTopConcept smls-skos:hasState ;
  skos:hasTopConcept smls-skos:isImplementedBy ;
  skos:hasTopConcept smls-skos:isRealizedBy ;
  skos:hasTopConcept smls-skos:hasRequirement ;
  skos:hasTopConcept smls-skos:hasRequirementSpecification ;
  skos:hasTopConcept smls-skos:hasRequirementSource ;
  skos:hasTopConcept smls-skos:hasRequirementSeverity ;
  .
smls-skos:Requirement
  a skos:Concept ;
  skos:broader smls-skos:InformationObject ;
  skos:related smls-skos:hasRequirementSpecification ;
  skos:related smls-skos:hasRequirementSource ;
  skos:related smls-skos:hasRequirementSeverity ;
  skos:definition "A rule that should hold for something, defined w.r.t. to
its attributes or relations" ;
  skos:example "The weight of a bridge should no exceed 100 tons"@en ;
```

```
    skos:prefLabel "Requirement"@en ;
.
smls-skos:RequirementSeverityType
  a skos:Concept ;
  skos:prefLabel "Requirement severity type"@en ;
.
smls-skos:RequirementSourceType
  a skos:Concept ;
  skos:prefLabel "Requirement source type"@en ;
.
smls-skos:hasRequirement
  a skos:Concept ;
  skos:prefLabel "has requirement"@en ;
.
smls-skos:hasRequirementSpecification
  a skos:Concept ;
  skos:prefLabel "has requirement specification"@en ;
.
smls-skos:hasRequirementSeverity
  a skos:Concept ;
  skos:prefLabel "has requirement severity"@en ;
.
smls-skos:hasRequirementSource
  a skos:Concept ;
  skos:prefLabel "has requirement source"@en ;
.
smls-skos:Activity
  a skos:Concept ;
  skos:broader smls-skos:Entity ;
  skos:definition "An activity is something possibly or actual happens in
space and time"@en ;
  skos:example "Fluid pressure measurement, driving a car are examples of an
<activity>"@en ;
  skos:prefLabel "Activity"@en ;
  skos:scopeNote "Process, function, human activity, machine activity, task,
procedure, project are subclasses of <Activity>"@en ;
.
smls-skos:Function
  a skos:Concept ;
  skos:broader smls-skos:Activity ;
  skos:definition "An activity performed by a functional object"@en ;
  skos:example "Pumping, connecting and producing are examples of a
<Function>"@en ;
  skos:prefLabel "Function"@en ;
.
smls-skos:Event
  a skos:Concept ;
  skos:definition "A thing that happens or takes place and marks the
beginning or ending of a state"@en ;
  skos:example "The connection of power to pump P_101, the take-off of a
aeroplane are examples of an <Event>"@en ;
  skos:prefLabel "Event"@en ;
  skos:scopeNote "Transition, cause, effect, risk, milestone, issue,
accident, point in time are subclasses of <Event>"@en ;
.
smls-skos:Object
  a skos:Concept ;
```

```
skos:broader smls-skos:Entity ;
skos:prefLabel "Object"@en ;
skos:scopeNote "<PhysicalObject> and <InformationObject> are subclasses of
<Object>"@en ;
.
smls-skos:Entity
  a skos:Concept ;
  skos:prefLabel "Entity"@en ;
  skos:scopeNote "<Object> and <Activity> are subclasses of <Entity>"@en ;
.
smls-skos:InformationObject
  a skos:Concept ;
  skos:broader smls-skos:Object ;
  skos:definition "Thing that is a whole of information on itself and has an
own identity"@en ;
  skos:example "A file within a computer system, the PDF file with URI
http://material-certificate/M-101-1234.pdf are examples of
<InformationObject>"@en ;
  skos:prefLabel "Information object"@en ;
  skos:scopeNote "Requirement, document, statement, E-mail, code, identifier
are subclasses of <InformationObject>"@en ;
.
smls-skos:PhysicalObject
  a skos:Concept ;
  skos:broader smls-skos:Object ;
  skos:definition "Is something that possibly or actually exists in space and
time, perceivable through the senses"@en ;
  skos:example "Pump P_101, a single living organism are examples of a
<PhysicalObject>"@en ;
  skos:prefLabel "Physical object"@en ;
  skos:scopeNote "Artefact, matter, person, organisation, stream, connection
are subclasses of <PhysicalObject>"@en ;
.
smls-skos:State
  a skos:Concept ;
  skos:definition "A particular condition that something is in during a
specific period in time"@en ;
  skos:example "Well-functioning of a car, the condition of an elevator of
being un-safe are examples of a <State>"@en ;
  skos:prefLabel "State"@en ;
  skos:scopeNote "Condition, failure state, objective are subclasses of
<State>"@en ;
.
smls-skos:begins
  a skos:Concept ;
  skos:related smls-skos:Event ;
  skos:related smls-skos:Activity ;
  skos:prefLabel "begins"@en ;
  skos:definition "This relationship applies between an Event and a
State."@en ;
  skos:example "The lightning strike (event) is the start of the forest fire
(state)"@en ;
.
smls-skos:PlannedObject
  a skos:Concept ;
  skos:broader smls-skos:PhysicalObject ;
  skos:prefLabel "Planned object"@en ;
```

```
    skos:definition "Is something that possibly (not actually) exists in space
and time"@en ;
.
smls-skos:BuiltObject
  a skos:Concept ;
  skos:broader smls-skos:PhysicalObject ;
  skos:prefLabel "Built object"@en ;
  skos:definition "Is something that actually exists in space and time"@en ;
.
smls-skos:FunctionalObject
  a skos:Concept ;
  skos:broader smls-skos:PhysicalObject ;
  skos:prefLabel "Functional object"@en ;
  skos:definition "Is something that possibly or actually exists in space and
time, from a functional perspective"@en ;
.
smls-skos:TechnicalObject
  a skos:Concept ;
  skos:broader smls-skos:PhysicalObject ;
  skos:prefLabel "Technical object"@en ;
  skos:definition "Is something that possibly or actually exists in space and
time, from a technical perspective"@en ;
.
smls-skos:describes
  a skos:Concept ;
  skos:related smls-skos:InformationObject ;
  skos:related smls-skos:PhysicalObject ;
  skos:related smls-skos:Activity ;
  skos:related smls-skos:Event ;
  skos:related smls-skos:State ;
  skos:related smls-skos:InformationObject ;
  skos:prefLabel "describes"@en ;
  skos:definition "This relationship applies between an Information Object
and a Physical Object, an Activity, an Event, a State, a Time Period, a
Spatial Region or another Information Object."@en ;
  skos:example "Take the Ketel bridge. Over time, a dossier has been created
of drawings, specifications, reports, invoices, in analogue or in digital
form. Each of these documents describes the Ketelbrug and can be seen as an
instance of an Information Object that describes the Ketelbrug"@en ;
.
smls-skos:ends
  a skos:Concept ;
  skos:related smls-skos:Event ;
  skos:related smls-skos:Activity ;
  skos:prefLabel "ends"@en ;
  skos:definition "This relationship applies between an Event and a
State."@en ;
  skos:example "The thunderstorm (event) ended the nice weather (state)"@en ;
.
smls-skos:hasState
  a skos:Concept ;
  skos:related smls-skos:PhysicalObject ;
  skos:related smls-skos:Activity ;
  skos:related smls-skos:State ;
  skos:prefLabel "has state"@en ;
  skos:definition "This relationship applies between an Activity and a State
or between a Physical Object and a State."@en ;
```

skos:example "A state can contain multiple parameters. In this way, the air pressure, temperature and humidity of a piece of atmosphere can be recorded. These three parameters then define the state of the atmosphere at a certain location and time"@en ;

.
smls-skos:performs
 a skos:Concept ;
 skos:related smls-skos:PhysicalObject ;
 skos:related smls-skos:Activity ;
 skos:prefLabel "performs"@en ;
 skos:definition "This relationship applies between a Physical Object and an Activity."@en ;
 skos:example "The welding of a bridge railing (an activity) is carried out with the aid of a welding machine (physical object) and the welder (physical object)"@en ;

.
smls-skos:transforms
 a skos:Concept ;
 skos:related smls-skos:Activity ;
 skos:related smls-skos:PhysicalObject ;
 skos:prefLabel "transforms"@en ;
 skos:definition "This relationship applies between an Activity and a Physical Object or between an Activity and an Information Object."@en ;
 skos:example "Baking (an activity) transforms a pizza (a physical object) from unbaked (a state) to baked (another state)"@en ;

.
smls-skos:triggers
 a skos:Concept ;
 skos:related smls-skos:PhysicalObject ;
 skos:related smls-skos:Activity ;
 skos:related smls-skos:Event ;
 skos:prefLabel "triggers"@en ;
 skos:definition "This relationship applies between an Event and an Activity and an Event and a Physical Object."@en ;
 skos:example "Take a traffic accident (an event). This initiates the arrival of the emergency services (an activity)"@en ;

.
smls-skos:isImplementedBy
 a skos:Concept ;
 skos:prefLabel "is implemented by"@en ;
 skos:related smls-skos:FunctionalObject ;
 skos:related smls-skos:TechnicalObject ;
 skos:definition "This relationship applies between a FunctionalObject and a TechnicalObject. The technical object fullfils or play the role of the functional object"@en ;

.
smls-skos:isRealizedBy
 a skos:Concept ;
 skos:prefLabel "is realized by"@en ;
 skos:related smls-skos:PlannedObject ;
 skos:related smls-skos:BuiltObject ;
 skos:definition "This relationship applies between a PlannedObject and a BuiltObject. The built object corresponds to the planned/designed imaginary object"@en ;

.
smls-skos:Demand
 a skos:Concept ;

```
    skos:prefLabel "Demand"@en ;
    skos:broader smls-skos:RequirementSeverityType ;
.
smls-skos:Wish
  a skos:Concept ;
  skos:prefLabel "Wish"@en ;
  skos:broader smls-skos:RequirementSeverityType ;
.
smls-skos:ByDefinition
  a skos:Concept ;
  skos:prefLabel "ByDefinition"@en ;
  skos:broader smls-skos:RequirementSourceType ;
.
smls-skos:ByClient
  a skos:Concept ;
  skos:prefLabel "ByClient"@en ;
  skos:broader smls-skos:RequirementSourceType ;
.
smls-skos:ByLawOrRegulation
  a skos:Concept ;
  skos:prefLabel "ByLawOrRegulation"@en ;
  skos:broader smls-skos:RequirementSourceType ;
.
smls-skos:ByRecommendation
  a skos:Concept ;
  skos:prefLabel "ByRecommendation"@en ;
  skos:broader smls-skos:RequirementSourceType ;
.
```

A.3 in RDFS (Turtle-Format)

```
# baseURI: https://w3id.org/def/smls-rdfs
# imports: http://qudt.org/schema/qudt
# imports: http://qudt.org/vocab/quantitykind
# imports: http://qudt.org/vocab/unit
# imports: http://www.w3.org/2004/02/skos/core
# prefix: smls-rdfs

@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .
@prefix qudt: <http://qudt.org/schema/qudt/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix smls-rdfs: <https://w3id.org/def/smls-rdfs#> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/def/smls-rdfs>
  rdf:type owl:Ontology ;
  owl:imports <http://qudt.org/schema/qudt> ;
  owl:imports <http://qudt.org/vocab/quantitykind> ;
  owl:imports <http://qudt.org/vocab/unit> ;
  owl:imports <http://www.w3.org/2004/02/skos/core> ;
.
smls-rdfs:Activity
  rdf:type rdfs:Class ;
```

```
    rdfs:subClassOf smls-rdfs:Entity ;
    skos:definition "An activity is something possibly or actual happens in
space and time"@en ;
    skos:example "Fluid pressure measurement, driving a car are examples of an
<activity>"@en ;
    skos:prefLabel "Activity"@en ;
    skos:scopeNote "Process, function, human activity, machine activity, task,
procedure, project are subclasses of <Activity>."@en ;
.
smls-rdfs:AddressSystem
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:SpatialReferenceSystem ;
    skos:prefLabel "Address system"@en ;
.
smls-rdfs:CalendarSystem
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:TemporalReferenceSystem ;
    skos:prefLabel "Calendar system"@en ;
.
smls-rdfs:CoordinateReferenceSystem
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:SpatialReferenceSystem ;
    skos:prefLabel "Coordinate reference system"@en ;
.
smls-rdfs:Entity
    rdf:type rdfs:Class ;
    skos:prefLabel "Entity"@en ;
    skos:scopeNote "<Object> and <Activity> are subclasses of <Entity>."@en ;
.
smls-rdfs:EnumerationType
    rdf:type rdfs:Class ;
    skos:definition "The superclass of all user-defined enumeration classes
where the allowed enumeration items are individuals"@en ;
    skos:example "LoadLevelType being Low, Medium or High"@en ;
    skos:prefLabel "Enumeration type"@en ;
.
smls-rdfs:Event
    rdf:type rdfs:Class ;
    skos:definition "A thing that happens or takes place and marks the
beginning or ending a state"@en ;
    skos:example "The connection of power to pump P_101, the take-off of a
aeroplane are examples of an <Event>"@en ;
    skos:prefLabel "Event"@en ;
    skos:scopeNote "Transition, cause, effect, risk, milestone, issue,
accident, point in time are subclasses of <Event>"@en ;
.
smls-rdfs:Function
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:Activity ;
    skos:definition "An activity performed by a functional object"@en ;
    skos:example "Pumping, connecting and producing are examples of a
<Function>"@en ;
    skos:prefLabel "Function"@en ;
.
smls-rdfs:FunctionalObject
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:PhysicalObject ;
    skos:prefLabel "Functional object"@en ;
```

```
    skos:definition "Is something that possibly or actually exists in space and
time, from a functional perspective"@en ;
.
smls-rdfs:PlannedObject
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:PhysicalObject ;
    skos:prefLabel "Planned object"@en ;
    skos:definition "Is something that possibly (not actually) exists in space
and time"@en ;
.
smls-rdfs:IndexKnownSpatialLocations
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:SpatialReferenceSystem ;
    skos:prefLabel "Index known spatial locations"@en ;
.
smls-rdfs:InformationObject
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:Object ;
    skos:definition "Thing that is a whole of information on itself and has an
own identity"@en ;
    skos:example "A file within a computer system, the PDF file with URI
http://material-certificate/M-101-1234.pdf are examples of
<InformationObject>"@en ;
    skos:prefLabel "Information object"@en ;
    skos:scopeNote "Requirement, document, statement, E-mail, code, identifier
are subclasses of <InformationObject>"@en ;
.
smls-rdfs:LinearReferenceSystem
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:SpatialReferenceSystem ;
    skos:prefLabel "Linear reference system"@en ;
.
smls-rdfs:Object
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:Entity ;
    skos:prefLabel "Object"@en ;
    skos:scopeNote "<PhysicalObject> and <InformationObject> are subclasses of
<Object>"@en ;
.
smls-rdfs:PhysicalObject
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:Object ;
    skos:definition "Is something that possibly or actually exists in space and
time, perceivable through the senses" ;
    skos:example "Pump P_101, a single living organism are examples of a
<PhysicalObject>"@en ;
    skos:prefLabel "Physical object"@en ;
    skos:scopeNote "Artefact, matter, person, organisation, stream, connection
are subclasses of <PhysicalObject>"@en ;
.
smls-rdfs:QuantityValue
    rdf:type rdfs:Class ;
    skos:prefLabel "Quantity value"@en ;
.
smls-rdfs:BuiltObject
    rdf:type rdfs:Class ;
    rdfs:subClassOf smls-rdfs:PhysicalObject ;
```

```
skos:prefLabel "Built object"@en ;
skos:definition "Is something that actually exists in space and time"@en ;
.
smls-rdfs:SpatialLocation
  rdf:type rdfs:Class ;
  skos:prefLabel "Spatial location"@en ;
.
smls-rdfs:SpatialReference
  rdf:type rdfs:Class ;
  skos:prefLabel "Spatial reference"@en ;
.
smls-rdfs:SpatialReferenceSystem
  rdf:type rdfs:Class ;
  skos:prefLabel "Spatial reference system"@en ;
.
smls-rdfs:SpatialRegion
  rdf:type rdfs:Class ;
  skos:definition "Demarcated space"@en ;
  skos:example "An IFC BREP or GeoSPARQL geometry representing a physical
space, as special case of a physical object, such as an office space, a
construction area for a building, a maritime traffic zone within the Channel,
a hazard zone within a plant or the loading gauge of a train are examples of
a <SpatialRegion>"@en ;
  skos:prefLabel "Spatial region"@en ;
.
smls-rdfs:State
  rdf:type rdfs:Class ;
  skos:definition "A particular condition that something is in during a
specific period in time"@en ;
  skos:example "Well-functioning of a car, the condition of an elevator of
being un-safe are examples of a <State>"@en ;
  skos:prefLabel "State"@en ;
  skos:scopeNote "Condition, failure state, objective are subclasses of
<State>"@en ;
.
smls-rdfs:TechnicalObject
  rdf:type rdfs:Class ;
  rdfs:subClassOf smls-rdfs:PhysicalObject ;
  skos:prefLabel "Technical object"@en ;
  skos:definition "Is something that possibly or actually exists in space and
time, from a technical perspective"@en ;
.
smls-rdfs:TemporalCoordinateReferenceSystem
  rdf:type rdfs:Class ;
  rdfs:subClassOf smls-rdfs:TemporalReferenceSystem ;
  skos:prefLabel "Temporal coordinate reference system"@en ;
.
smls-rdfs:TemporalLocation
  rdf:type rdfs:Class ;
  skos:prefLabel "Temporal location"@en ;
.
smls-rdfs:TemporalReference
  rdf:type rdfs:Class ;
  skos:prefLabel "Temporal reference"@en ;
.
smls-rdfs:TemporalReferenceSystem
  rdf:type rdfs:Class ;
```

```
    skos:prefLabel "Temporal reference system"@en ;
.
smls-rdfs:TemporalRegion
  rdf:type rdfs:Class ;
  skos:definition "A length or portion of time"@en ;
  skos:example "Week 12 in 2019, design stage of the Schiphol tunnel are
examples of a <TemporalRegion>"@en ;
  skos:prefLabel "Temporal region"@en ;
  skos:scopeNote "Week, day, life cycle stage are subclasses of
<TemporalRegion>"@en ;
.
smls-rdfs:Requirement
  a rdfs:Class ;
  rdfs:subClassOf smls-rdfs:InformationObject ;
  skos:definition "A rule that should hold for something, defined w.r.t. to
its attributes or relations" ;
  skos:example "The weight of a bridge should no exceed 100 tons"@en ;
  skos:prefLabel "Requirement"@en ;
.
smls-rdfs:RequirementSeverityType
  a rdfs:Class ;
  rdfs:subClassOf smls-rdfs:EnumerationType ;
  skos:prefLabel "Requirement severity type"@en ;
.
smls-rdfs:Wish
  a smls-rdfs:RequirementSeverityType ;
.
smls-rdfs:Demand
  a smls-rdfs:RequirementSeverityType ;
.
smls-rdfs:RequirementSourceType
  a rdfs:Class ;
  rdfs:subClassOf smls-rdfs:EnumerationType ;
  skos:prefLabel "Requirement source type"@en ;
.
smls-rdfs:ByClient
  a smls-rdfs:RequirementSourceType ;
.
smls-rdfs:ByDefinition
  a smls-rdfs:RequirementSourceType ;
.
smls-rdfs:ByLawOrRegulation
  a smls-rdfs:RequirementSourceType ;
.
smls-rdfs:ByRecommendation
  a smls-rdfs:RequirementSourceType ;
.
smls-rdfs:hasRequirement
  a rdf:Property ;
  rdfs:range smls-rdfs:Requirement ;
  skos:prefLabel "has requirement"@en ;
.
smls-rdfs:hasRequirementSpecification
  a owl:DatatypeProperty ;
  rdfs:domain smls-rdfs:Requirement ;
  rdfs:range xsd:string;
  skos:prefLabel "has requirement specification"@en ;
```

```
.
smls-rdfs:hasRequirementSeverity
  a rdf:Property ;
  rdfs:domain smls-rdfs:Requirement ;
  rdfs:range smls-rdfs:RequirementSeverityType ;
  skos:prefLabel "has requirement severity"@en ;

.
smls-rdfs:hasRequirementSource
  a rdf:Property ;
  rdfs:domain smls-rdfs:Requirement ;
  rdfs:range smls-rdfs:RequirementSourceType ;
  skos:prefLabel "has requirement source"@en ;

.
smls-rdfs:abbreviation
  rdf:type rdf:Property ;
  rdfs:subPropertyOf skos:altLabel ;
  skos:prefLabel "abbreviation"@en ;

.
smls-rdfs:begins
  rdf:type rdf:Property ;
  rdfs:domain smls-rdfs:Event ;
  rdfs:range smls-rdfs:State ;
  skos:definition "This relationship applies between an Event and a State"@en
;
  skos:example "The lightning strike (event) is the start of the forest fire
(state)"@en ;
  skos:prefLabel "begins"@en ;

.
smls-rdfs:describes
  rdf:type rdf:Property ;
  rdfs:domain smls-rdfs:InformationObject ;
  skos:definition "This relationship applies between an Information Object
and a Physical Object, an Activity, an Event, a State, a Time Period, a
Spatial Region or another Information Object"@en ;
  skos:example "Take the Ketel bridge. Over time, a dossier has been created
of drawings, specifications, reports, invoices, in analogue or in digital
form. Each of these documents describes the Ketelbrug and can be seen as an
instance of an Information Object that describes the Ketelbrug"@en ;
  skos:prefLabel "describes"@en ;

.
smls-rdfs:ends
  rdf:type rdf:Property ;
  rdfs:domain smls-rdfs:Event ;
  rdfs:range smls-rdfs:State ;
  skos:definition "This relationship applies between an Event and a State"@en
;
  skos:example "The thunderstorm (event) ended the nice weather (state)"@en ;
  skos:prefLabel "ends"@en ;

.
smls-rdfs:hasBoundary
  rdf:type rdf:Property ;
  rdfs:range smls-rdfs:SpatialRegion ;
  skos:definition "This relationship applies between a Physical Object and a
Spatial Region"@en ;
  skos:example "A parcel is bounded by a 1D GML polyline"@en ;
  skos:prefLabel "has boundary"@en ;

.
```