

# SVENSK STANDARD

## SS-EN 17632-1:2022

**Byggnadsinformationsmodellering (BIM) – Semantisk modellering och länkning (SML) – Del 1: Generiska modelleringsmönster**

**Building information modelling (BIM) – Semantic modelling and linking (SML) – Part 1: Generic modelling patterns**



**Sis** Svenska  
Institutet för  
Standarder

Språk: engelska/English

Utgåva: 1

**SIS multi user license: Trafikverket. Customer number: , Date:  
2023-02-23**



---

Den här standarden kan hjälpa dig att effektivisera och kvalitetssäkra ditt arbete. SIS har fler tjänster att erbjuda dig för att underlätta tillämpningen av standarder i din verksamhet.

#### **SIS Abonnemang**

Snabb och enkel åtkomst till gällande standard med SIS Abonnemang, en prenumerationstjänst genom vilken din organisation får tillgång till all världens standarder, senaste uppdateringarna och där hela din organisation kan ta del av innehållet i prenumerationen.

#### **Utbildning, event och publikationer**

Vi erbjuder även utbildningar, rådgivning och event kring våra mest sålda standarder och frågor kopplade till utveckling av standarder. Vi ger också ut handböcker som underlättar ditt arbete med att använda en specifik standard.

#### **Vill du delta i ett standardiseringsprojekt?**

Genom att delta som expert i någon av SIS 300 tekniska kommittéer inom CEN (europeisk standardisering) och/eller ISO (international standardisering) har du möjlighet att påverka standardiseringsarbetet i frågor som är viktiga för din organisation. Välkommen att kontakta SIS för att få veta mer!

#### **Kontakt**

Skriv till [kundservice@sis.se](mailto:kundservice@sis.se), besök [sis.se](http://sis.se) eller ring 08 - 555 523 10

---

© Copyright/Upphovsrätten till denna produkt tillhör Svenska institutet för standarder, Stockholm, Sverige. Upphovsrätten och användningen av denna produkt regleras i slutanvändarlicensen som återfinns på [sis.se/slutanvendarlicens](http://sis.se/slutanvendarlicens) och som du automatiskt blir bunden av när du använder produkten. För ordlista och förkortningar se [sis.se/ordlista](http://sis.se/ordlista).

© Copyright Svenska institutet för standarder, Stockholm, Sweden. All rights reserved. The copyright and use of this product is governed by the end-user licence agreement which you automatically will be bound to when using the product. You will find the licence at [sis.se/enduserlicenseagreement](http://sis.se/enduserlicenseagreement).

Upplysningar om sakinnehållet i standarden lämnas av Svenska institutet för standarder, telefon 08 - 555 520 00. Standarder kan beställas hos SIS som även lämnar allmänna upplysningar om svensk och utländsk standard.

Standarden är framtagen av kommittén för Bygg- och förvaltningsdokumentation, SIS/TK 269.

Har du synpunkter på innehållet i den här standarden, vill du delta i ett kommande revideringsarbete eller vara med och ta fram andra standarder inom området? Gå in på [www.sis.se](http://www.sis.se) - där hittar du mer information.



---

Europastandarden EN 17632-1:2022 gäller som svensk standard. Detta dokument innehåller den officiella engelska versionen av EN 17632-1:2022.

The European Standard EN 17632-1:2022 has the status of a Swedish Standard. This document contains the official version of EN 17632-1:2022.



## LÄSANVISNINGAR FÖR STANDARDER

I dessa anvisningar behandlas huvudprinciperna för hur regler och yttrre begränsningar anges i standardiseringsprodukter.

### Krav

Ett krav är ett uttryck i ett dokuments innehåll som anger objektivt verifierbara kriterier som ska uppfyllas och från vilka ingen avvikelse tillåts om efterlevnad av dokumentet ska kunna åberopas. Krav uttrycks med hjälpverbet **ska** (eller **ska inte** för förbud).

### Rekommendation

En rekommendation är ett uttryck i ett dokuments innehåll som anger en valmöjlighet eller ett tillvägagångssätt som bedöms vara särskilt lämpligt utan att nödvändigtvis nämna eller utesluta andra. Rekommendationer uttrycks med hjälpverbet **bör** (eller **bör inte** för avrådanden).

### Instruktion

Instruktioner anges i imperativ form och används för att ange hur något görs eller utförs. De kan underordnas en annan regel, såsom ett krav eller en rekommendation. De kan även användas självständigt, och är då att betrakta som krav.

### Förklaring

En förklaring är ett uttryck i ett dokuments innehåll som förmedlar information. En förklaring kan uttrycka tillåtelse, möjlighet eller förmåga. Tillåtelse uttrycks med hjälpverbet **får**. Inom standardiseringen saknas rekommenderad nekande motsats till hjälpverbet **får**, förbud uttrycks med **ska inte** enligt reglerna för krav. Möjlighet och förmåga uttrycks med hjälpverbet **kan** (eller motsatsen **kan inte**).

## READING INSTRUCTIONS FOR STANDARDS

These instructions cover the main principles for the use of provisions and external constraints in standardization deliverables.

### Requirement

A requirement is an expression, in the content of a document, that conveys objectively verifiable criteria to be fulfilled, and from which no deviation is permitted if conformance with the document is to be claimed. Requirements are expressed by the auxiliary **shall** (or **shall not** for prohibition).

### Recommendation

A recommendation is an expression, in the content of a document, that conveys a suggested possible choice or course of action deemed to be particularly suitable, without necessarily mentioning or excluding others. Recommendations are expressed by the auxiliary **should** (or **should not** for dissuasion).

### Instruction

An instruction is expressed in the imperative mood and is used in order to convey an action to be performed. It can be subordinated to another provision, such as a requirement or a recommendation. It can also be used independently and is then to be regarded as a requirement.

### Statement

A statement is an expression, in the content of a document, that conveys information. A statement can express permission, possibility or capability. Permission is expressed by the auxiliary **may**. There is no recommended opposite expression for the auxiliary **may** in standardization, prohibition is expressed by the use of **shall not** in accordance with the rules for requirements. Possibility and capability are expressed by the auxiliary **can** (its opposite being **cannot**).

EUROPEAN STANDARD

**EN 17632-1**

NORME EUROPÉENNE

EUROPÄISCHE NORM

December 2022

ICS 35.240.67

English Version

**Building information modelling (BIM) - Semantic  
modelling and linking (SML) - Part 1: Generic modelling  
patterns**

Modélisation d'informations de la construction (BIM) -  
Modélisation et liaisons sémantiques (SML) - Partie 1 :  
Schémas de modélisation génériques

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

This European Standard was approved by CEN on 12 September 2022.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION  
COMITÉ EUROPÉEN DE NORMALISATION  
EUROPÄISCHES KOMITEE FÜR NORMUNG

**CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels**

## SS-EN 17632-1:2022 (E)

**Contents**

	Page
<b>European foreword.....</b>	<b>4</b>
<b>Introduction .....</b>	<b>5</b>
<b>1 Scope.....</b>	<b>6</b>
<b>2 Normative references.....</b>	<b>7</b>
<b>3 Terms and definitions .....</b>	<b>8</b>
<b>4 Symbols and abbreviated terms.....</b>	<b>11</b>
<b>4.1 Symbols.....</b>	<b>11</b>
<b>4.2 Abbreviated terms.....</b>	<b>11</b>
<b>5 Semantic modelling levels of capability.....</b>	<b>13</b>
<b>6 L1: Information language.....</b>	<b>14</b>
<b>6.1 Conceptual L1: Information language.....</b>	<b>14</b>
<b>6.2 Concrete L1: Information language bindings.....</b>	<b>16</b>
<b>6.3 Modelling patterns .....</b>	<b>19</b>
<b>7 M1: Information model.....</b>	<b>28</b>
<b>7.1 Top level information model .....</b>	<b>28</b>
<b>7.2 Systems engineering extension.....</b>	<b>30</b>
<b>8 Implementing SML in code.....</b>	<b>32</b>
<b>9 Conformance.....</b>	<b>32</b>
<b>9.1 General.....</b>	<b>32</b>
<b>9.2 Conformance on language level.....</b>	<b>32</b>
<b>9.3 Conformance on semantic level.....</b>	<b>33</b>
<b>Annex A (normative) SML implementation in 'linked data'.....</b>	<b>34</b>
<b>A.1 Introduction.....</b>	<b>34</b>
<b>A.2 SKOS part.....</b>	<b>34</b>
<b>A.3 RDFS part .....</b>	<b>40</b>
<b>A.4 OWL part .....</b>	<b>48</b>
<b>A.5 SHACL part.....</b>	<b>53</b>
<b>Annex B (normative) Selected W3C RDF language subsets .....</b>	<b>58</b>
<b>B.1 General.....</b>	<b>58</b>
<b>B.2 XML schema (XSD), part 2: Datatypes 2<sup>nd</sup> edition .....</b>	<b>58</b>
<b>B.3 Resource description framework (RDF).....</b>	<b>58</b>
<b>B.4 Simple knowledge organization system (SKOS) .....</b>	<b>59</b>
<b>B.5 Resource description framework schema (RDFS) .....</b>	<b>59</b>
<b>B.6 Web ontology language (OWL).....</b>	<b>60</b>
<b>B.7 Shape constraint language (SHACL) .....</b>	<b>61</b>
<b>Annex C (informative) SML Example in SKOS/RDFS/OWL/SHACL (Turtle format).....</b>	<b>64</b>
<b>C.1 Example description .....</b>	<b>64</b>
<b>C.2 SKOS part.....</b>	<b>64</b>

<b>C.3</b>	<b>RDFS part.....</b>	<b>66</b>
<b>C.4</b>	<b>OWL part.....</b>	<b>69</b>
<b>C.5</b>	<b>SHACL part .....</b>	<b>70</b>
<b>C.6</b>	<b>Data part .....</b>	<b>71</b>
<b>Annex D (informative) Relationships with other asset/product modelling standards .....</b>		<b>73</b>
<b>D.1</b>	<b>General .....</b>	<b>73</b>
<b>D.2</b>	<b>Relationship with the ISO 21597 series .....</b>	<b>73</b>
<b>D.3</b>	<b>Relationship with EN ISO 23387 .....</b>	<b>73</b>
<b>D.4</b>	<b>Relationship with the ISO 15926 series .....</b>	<b>92</b>
<b>Annex E (informative) Linking information .....</b>		<b>94</b>
<b>E.1</b>	<b>Types of linking.....</b>	<b>94</b>
<b>E.2</b>	<b>Language-level language link sets.....</b>	<b>94</b>
<b>Bibliography .....</b>		<b>96</b>

## **SS-EN 17632-1:2022 (E)**

### **European foreword**

This document (EN 17632-1:2022) has been prepared by Technical Committee CEN/TC 422 “Building information modelling (BIM)”, the secretariat of which is held by SN - Norway.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by June 2023, and conflicting national standards shall be withdrawn at the latest by June 2023.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

Any feedback and questions on this document should be directed to the users' national standards body. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organisations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

## Introduction

This document is about the built environment. In the built environment, assets relating to buildings and infrastructures need to be managed across their entire life cycle, involving programming, design, construction, operation, modification and demolition or disassembly. Vast amounts of valuable information about them are created or captured, stored and communicated according to a diverse range of forms and structures - and often lost again.

To manage these projects and their resulting assets more efficiently and effectively, information needs to be findable, accessible, interoperable and reusable (FAIR). The world wide web consortium (W3C) provides open and generic linked data (LD) and semantic web (SW) technologies [1] which are capable of providing this 'FAIRness' giving information a common form ('syntax') and structure ('semantics').

Using the 'new European Interoperability Framework' (EIF) [9] terminology, this document focuses on syntactic and semantic interoperability.

This document specifies how organizations in the built environment can apply this W3C technology to best suit their needs. For example, it can be used within organizations to communicate information internally between various business departments and software, or it can be used externally to publish information across the multitude of databases and organizations in the sector.

Application of this document will in particular help to align and integrate relevant 'modelling worlds' for the built environment, typically involving already existing complex information models, like in Building Information Modelling (BIM), Geographical Information Systems (GIS), Systems Engineering (SE), Monitoring & Control (M&C) and Electronic Document Management (EDM).

Regarding to BIM Building Information Modelling, this document has been prepared with the EN ISO 16739-1 [11] Industry Foundation Classes (IFC) information model in mind, and it has been aligned with the revision work of EN ISO 12006-3 [17] (used to extend IFC via a buildingSmart data dictionary (bSDD)). More specifically, this document offers a 'linked data' view on the 'data templates' related to CEN TC442/WG4. It provides a way to represent the 'attributes' for 'properties' of EN ISO 23386:2020 [15] implemented according to EN ISO 23387:2020 [16], again involving EN ISO 12006-3.

As any other technical specification, this document requires expertise and experience in specifying, procuring and delivering work results. As semantic modelling and linking is in the domain of computer science, the content is aimed at those professionals. This document however, provides a standardized approach for the built environment, and thus this introduction addresses the sector and its decision makers.

Wherever the sector could benefit from better ways of searching, finding and (re)using information, this document specifies how to store, model, publish and link this information, with the aim of modelling information once in a standardized way, instead of adapting and transforming information on an ad hoc basis. In other words, it is not a matter of shifting information structures already in place, but a matter of modelling them for publishing on the Web/internet in more cloud-native ways.

The key principle of this document is to keep semantic modelling as simple and standardized as possible. The objectives for capability range from machine-readable information (interpreted by humans) via (as far as possible) machine-interpretable information to fully integrated and interlinked information sources.

This document is complementary to other ISO standards. In the Annex D, related ISO standards are listed and the exact relationships are described.

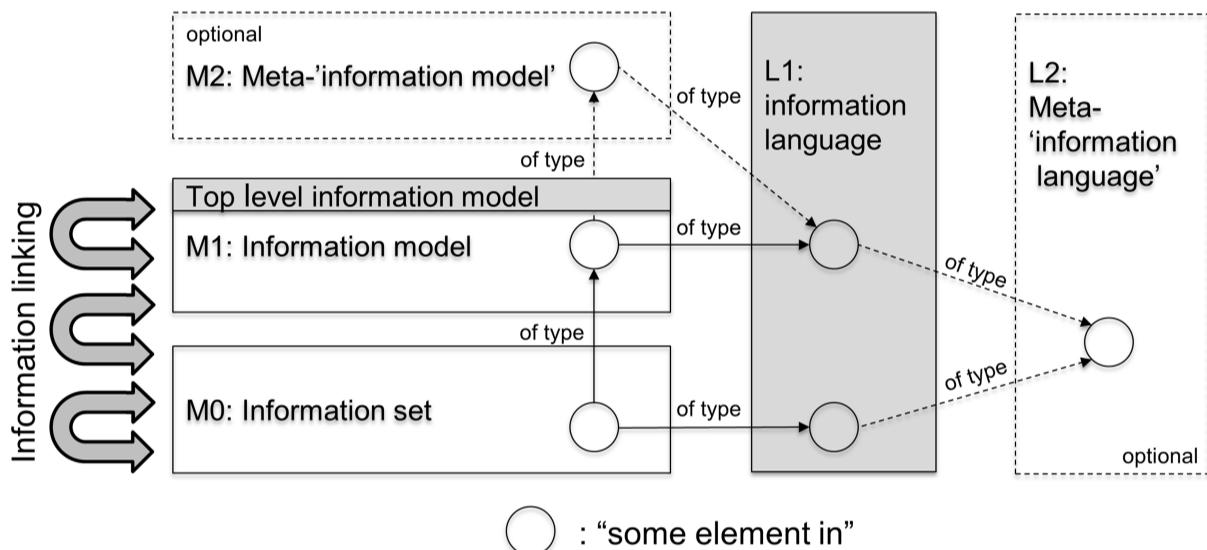
The standardized modelling patterns introduced in this document may be applicable to other industry sectors as well.

## SS-EN 17632-1:2022 (E)

### 1 Scope

This document addresses *syntactic and semantic interoperability* for information describing assets going through their life cycle in the built environment. It assumes the underlying *technical interoperability* provided already by the Internet/World Wide Web (WWW) technology-stack. The syntactic aspects relate to the Linked Data (LD)/Semantic Web (SW) formats and the SPARQL direct access method provided. The semantic aspects relate to the LD/SW-based information models in the form of thesauri and ontologies giving meaning to the information.

The following information architecture (Figure 1) applies.



**Figure 1 — Information architecture with (grey areas indicating the scope of this document)**

This document specifies:

- a conceptual “L1: Information language” with four RDF-based language bindings being SKOS, RDFS, OWL and SHACL, including:
  - a choice of ‘linked data’/RDF-based formats (to be used for all modelling and language levels); and
- a generic Top Level Information Model of a total “M1: Information model”, here “an upper ontology”, including:
  - a set of generic information modelling patterns for identification, annotation, enumeration datatypes, complex quality/quantity modelling, decomposition and grouping.

This modelling approach for information models and information sets is relevant within the built environment from multiple perspectives such as:

- Building information modelling (BIM);
- Geographical information systems (GIS);
- Systems engineering (SE);
- Monitoring & control (M&C); and
- Electronic document management (EDM).

Annex E discusses in an informative way how the information models and sets relevant for these different worlds can be linked together using LD/SW technology.

This document does not specify a full meta-'information model', sometimes referred to as a 'Knowledge Model (KM)'. EN ISO 12006-3 provides such an often used model for the built environment. In Annex D, Subclause D.3 it is shown how this existing model can be made compliant to this document. The only direct support for this meta level comes in the form of the possibility to define 'types' (enumeration types or concept types) and 'objectifications' as metaconcepts.

This document does not specify a meta-'information language' since this is already provided by the concrete RDF-based language bindings (being RDFS).

The scope of this document in general excludes the following:

- Business process modelling;
- Software implementation aspects;
- Information packaging and transportation/transaction aspects already handled by ISO TC59/SC13 Information container for linked document delivery (ICDD) ([13]) respectively various information delivery manual (IDM) / information exchange requirements (EIR)-related initiatives; and
- Domain-specific (here: 'built environment'-specific) content modelling in the form of concepts, attributes and relations at end-user level (the actual ontologies themselves) beyond a generic top level information model ('upper ontology') and modelling and linking patterns.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements for this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

JSON-LD 1.1, A JSON-based Serialization for Linked Data, W3C Recommendation, 16 July 2020, <https://www.w3.org/TR/json-ld11/>

OWL 2<sup>1</sup> 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/>

---

<sup>1</sup> Hereafter referred to as just "OWL".

## SS-EN 17632-1:2022 (E)

SPARQL 1.1 Overview, 21 March 2013, W3C Recommendation, <https://www.w3.org/TR/sparql11-overview/> (referencing, among others, the next two, more specific, references)

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

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 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 6707-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

#### 3.1

##### **asset**

item, thing or entity that has potential or actual value to an organization

[SOURCE: ISO 55000:2014, 3.2.1, modified – Note 1, 2 and 3 to entry have been removed.]

#### 3.2

##### **attribute**

inherent characteristic

Note 1 to entry: The term used in EN ISO 12006-3 is xtdProperty.

[SOURCE: EN ISO 9241-302:2008, 3.4.2, modified – Note 1 to entry has been added.]

#### 3.3

##### **built environment**

collection of man-made or induced physical objects located in a particular area or region

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

#### 3.4

##### **closed-world assumption**

##### **CWA**

assumption, in a formal system of logic used for knowledge representation that a statement that is true is also known to be true; therefore, conversely, what is not currently known to be true is false

Note 1 to entry: Typically combined with the Unique Name Assumption (UNA).

#### 3.5

##### **concept**

abstract entity for determining category membership

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

**3.6**

**data format**

predetermined arrangement of data on a data medium

[SOURCE: ISO 5127:2017, 3.1.13.12]

**3.7**

**exchange information requirement**

**EIR**

information requirement in relation to an appointment

[SOURCE: EN ISO 19650-1:2018, 3.3.6]

**3.8**

**hierarchy**

concept system in which all concepts are related in hierarchical relations that form a partial ordering

[SOURCE: ISO/IEC TR 11179-2:2019, 3.8]

**3.9**

**information model**

**data model**

description of the organization of information giving structure/meaning ('semantics') to an information set

**3.10**

**information set**

**data set**

named collection of information describing or specifying something you can or could point at in reality

**3.11**

**level of capability**

**LoC**

semantic modelling power provided by the 'linked data' languages related to the needs of a specific use case type

**3.12**

**machine-interpretable**

able to (to a certain extent) be semantically interpreted by a computer

**3.13**

**machine-readable**

able to be read and processed by a computer

**3.14**

**meronymy**

type of hierarchy which deals with part-whole relationships

[SOURCE: ISO/IEC 11179-3:2013, 3.2.73]

**3.15**

**metadata**

data about data (documents, information sets, information models or elements in those)

## **SS-EN 17632-1:2022 (E)**

### **3.16**

#### **n-ary**

having an arity of n

### **3.17**

#### **object**

any part of the perceivable or conceivable world

Note 1 to entry: An object is something abstract or physical toward which thought, feeling, or action is directed.

Note 2 to entry: Within this document, the term individual is used as synonym of object.

[SOURCE: EN ISO 12006-2:2020, 3.1.1, modified – added Note 1 and Note 2 to entry.]

### **3.18**

#### **ontology**

formal, explicit specification of a shared conceptualization

Note 1 to entry: An ontology typically includes definitions of concepts and specified relationships between them, set out in a formal way so that a machine can use them for reasoning.

[SOURCE: ISO 25964-2:2013, definition 3.57]

Note 2 to entry: See also ISO/TR 13054:2012, definition 2.6; ISO/TS 13399-4:2014, definition 3.20; ISO 19101-1:2014, definition 4.1.26; ISO 18435-3:2015, definition 3.1; ISO/IEC 19763-3:2010, definition 3.1.1.1.

Note 3 to entry: Applied in this document as a set of concepts, reference individuals, value types, reference values, attributes, relations, constraints and derivations.

[SOURCE: ISO 5127:2017, 3.1.2.03, modified – added Note 3 to entry.]

### **3.19**

#### **open-world assumption**

#### **OWA**

opposite of the closed-world assumption stating that lack of knowledge does not imply falsity

Note 1 to entry: typically combined with the No Unique Name Assumption (NO-UNA).

### **3.20**

#### **property**

attribute or a relation

Note 1 to entry: This is also the term used in RDF (rdf:Property).

### **3.21**

#### **relation**

#### **relationship**

sense in which concepts can be connected, via constituent roles

Note 1 to entry: The related concepts may be general or individual concepts.

Note 2 to entry: The term used in EN ISO 12006-3 is xtdRelationshipToSubject.

EXAMPLE      Causality is a relation with two constituent roles: cause and effect.

[SOURCE: ISO/IEC 11179-3:2013, 3.2.119, modified – added Note 2 to entry.]

**3.22****systems engineering****SE**

interdisciplinary approach governing the total technical and managerial effort required to transform a set of stakeholder needs, expectations, and constraints into a solution and to support that solution throughout its life

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

**3.23****taxonomy**

type of hierarchy which deals with generalization/specialization relationships

[SOURCE: ISO/IEC 11179-3:2013, 3.2.135]

**3.24****top level information model**

generic part of an information model (typically a generic taxonomy)

**3.25****triple**

statement in the form subject-predicate-object that expresses a fact

**3.26****typology**

type of hierarchy which deals with classification/instantiation relationships

**3.27****use case**

sequence of actions that an actor (usually a person, but perhaps an external entity, such as another system) performs within a system to achieve a particular goal

[SOURCE: ISO/TR 17185-3:2015, 3.17; ISO/TR 25102, modified]

## **4 Symbols and abbreviated terms**

### **4.1 Symbols**

This document does not contain any symbols.

### **4.2 Abbreviated terms**

For the purposes of this document, the following abbreviated terms apply.

API	application programming interface
BIM	building information modelling
bSDD	buildingSmart data dictionary
CWA	closed world assumption
ECMA	European computer manufacturers association international
EDM	electronic document management
EIF	European interoperability framework

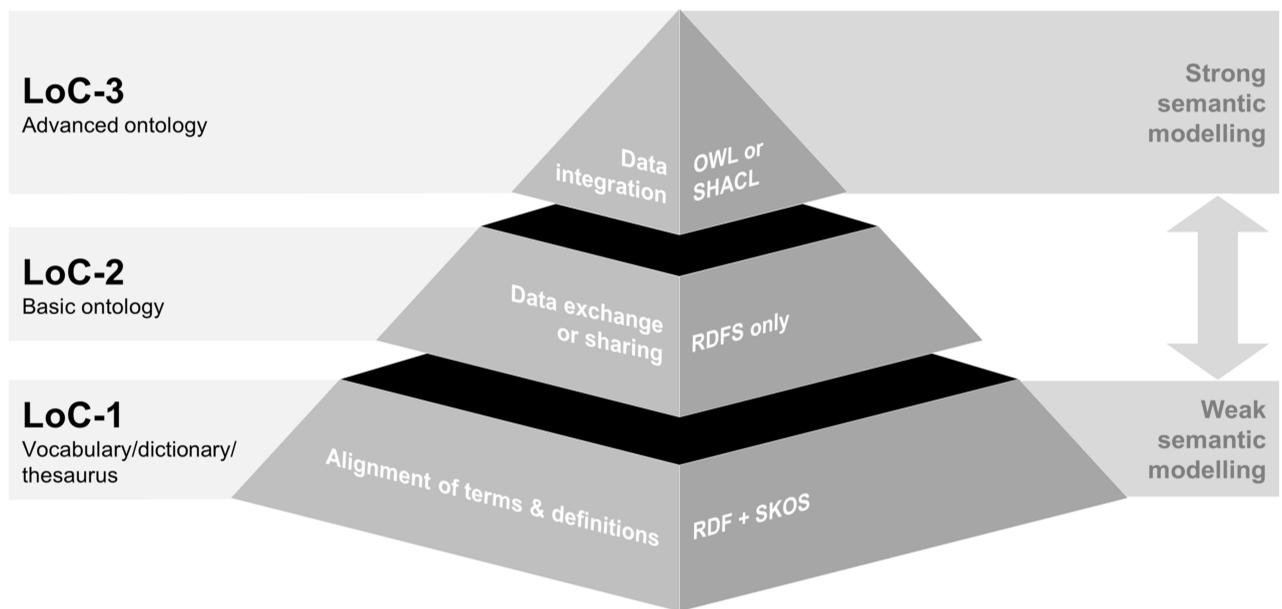
**SS-EN 17632-1:2022 (E)**

EIR	exchange information requirements
FAIR	findable, accessible, interoperable, reusable [go-fair.org]
GIS	geographical information systems
GUID	globally unique identifier (typically assigned)
ICDD	information container for linked document delivery [ISO]
IDM	information delivery manual
IFC	industry foundation classes [ISO]
IETF	internet engineering task force
JSON	JavaScript object notation [ECMA]
JSON-LD	JavaScript object notation - linked data [W3C]
LD	linked data (technology) [W3C]
LoC	level of capability
M&C	monitoring & control
OMG	object management group
OWA	open world assumption
OWL	web ontology language [W3C]
QUDT	quantities, units, dimensions and data types [qudt.org]
RDF	resource description framework [W3C]
RDFS	resource description framework schema [W3C]
RFC	request for comments [IETF]
SE	systems engineering
SHACL	shapes constraints language [W3C]
SML	semantic modelling and linking [CEN]
SPARQL	sparql protocol and RDF query language [W3C]
SPFF	step physical file format [STEP]
SSoF	single source of facts
STEP	standard for the exchange of product model data [ISO]
SW	semantic web (technology) [W3C]
UML	unified modelling language [OMG]
URI	uniform resource identifier [W3C]
UUID	universally unique identifier [IETF]
XML	extensible markup language [W3C]
XSD	extensible markup language schema definition [W3C]
W3C	world wide web consortium
WWW	world wide web [W3C]

## 5 Semantic modelling levels of capability

An appointing party shall define the levels of capability required for each use case.

Different use case types need different 'levels of (semantic modelling) capability' (LoC) related to the required modelling power. This document specifies three main LoCs (Figure 2).



**Figure 2 — Three use case types and related 'Levels of (semantic modelling) Capability' (LoCs)**

The left part of Figure 2 represents the organizational use case type activity, the right side the related (linked data) modelling languages available. The simplest use case type, requiring the weakest semantic modelling, is the common understanding and *alignment of terms and definitions* used to describe assets, their environment and internal structure. Weak modelling is sufficient here as a first step especially targeted towards human interpretation. A good definition gives an end user guidance on how to later classify and instantiate their information according to these terms. This level targets common human understanding of terms and definitions and at least making sure the information is machine-readable. Terms and definitions can be distributed as well as published on websites for others to refer to and reuse. For this LoC-1 the RDF framework and the SKOS language shall be used.

More expressive power is needed whenever common understanding is required for the *exchange or sharing* of asset information between digital systems (within or between parties). This can be achieved with LoC-2 where information is classified according to basic ontologies involving concepts, valuetypes, attributes and relations and restrictions. This stronger level builds upon the lowest level making the information also machine-interpretable. Because of this adding of semantics, limited automatic inference of information from asserted information becomes possible. For this LoC-2 the RDF framework and the RDFS language shall be used.

NOTE 1 OWL is defined on top of RDFS and is providing more modelling power than RDFS by introducing 'restrictions' that can be used to infer new information from asserted information.

The term 'machine-interpretable' should be understood in an informal sense.

NOTE 2 Machine-interpretation like humans do would ultimately require understanding of all the names and labels used for the concepts, attributes and relations. At least the ontologies provide a level of structure defining the possibilities and impossibilities for the data provided giving some machine-interpretable meaning to the information.

## SS-EN 17632-1:2022 (E)

Finally, to be able to *integrate* data for all kinds of decision support, even stronger semantics are needed. This stronger knowledge comes in the form of explicit constraints and rules for the data against which data can be verified respectively inferred assuming an open-world assumption (OWA) or closed-world assumption (CWA). These constraints and rules can be definitional or operational, the latter specified by the client brief or legalisation/regulation body on top of the definitional ones. For this LoC-3 the RDF framework and (RDFS+OWL) and/or (RDFS+SHACL) shall be used.

Note 3 The W3C Recommendation for the SHACL specification (consisting of two parts: SHACL Core and SHACL-SPARQL) is used for data verification, the SHACL 1.1 Advanced Features draft community track report [5] for data inference.

Note 4 The default interpretation for data inference in case of OWL modelling is OWA, the default interpretation for data verification in case of SHACL modelling is CWA.

## 6 L1: Information language

### 6.1 Conceptual L1: Information language

The conceptual L1: Information language that is still independent from concrete RDF-based languages is specified as in Table 1.

**Table 1 — Conceptual L1: Information language**

Meta-set		
<b>1. Information model</b>		
<b>2. Information set</b>		
<b>3. Group</b>		
Meta-concept		
<b>4. Concept</b>		
<b>5. Individual</b>		
<b>6. Value type</b>		
<b>7. Value</b>		
<b>8. Attribute</b>		
<b>8.1 Annotation</b>		
<b>8.2 Quality</b>		
<b>8.3 Quantity</b>		
<b>9. Relation</b>		
Meta-relation	From <sup>2</sup>	To
10. Grouping	1 to 3 (or)	3 to 9 (or)
11. Classification (inverse: Instantiation)	4 or 5	4
	7	6
12. Generalization (inverse: Specialization)	4	4
	6	6
	8	8
	9	9
13. Constraint	(more complex n-ary)	
14. Derivation	(more complex n-ary)	

<sup>2</sup> From/To Meta-set/Meta-concept.

This conceptual information language contains meta-sets, meta-concepts and meta-relations.

## Meta-sets

The first meta-set reflects a specification for information sets in general referred to as an information model. Good examples of information models are vocabularies, thesauri, dictionaries and (basic or advanced) ontologies. View models that only reuse (without changing) existing other models are also an example of information models. In addition, there are information sets that contain individual information typically according to/controlled by some information model. Finally, there can be user-defined groups reflecting subsets of the information model or information set.

## Meta-concepts

The first basic meta-concept being a member of an information model is a *concept* referring to abstract notions as type of things of interest. The next one is an *individual*, an instance of a concept representing something you can or could point at in reality. The next meta-concept is a *valuetype*, like a string, decimal, double, boolean or enumeration type that have *values* as instances.

Then there are *attributes* describing intrinsic characteristics and *relations* describing extrinsic characteristics of individuals of concepts. Attributes are further divided in *annotations*, adding human-interpretable identifiers, names, labels, definitions etc. and machine-interpretable *qualities* and *quantities*.

Relations in this document shall be binary relations since all specified language bindings in this document use binary relations only. In case there is a need to model n-ary relations ( $n > 2$ ) the modelling approach shall be used as described in [6] following the specific complex relation pattern of 6.3.9.

- An example relation is decomposition (see 6.3.7), where constraints on this decomposition define a socalled ‘meronomy’ or ‘typical decomposition’.

## Meta-relations

Two meta-relations are specified:

- Classification (inverse: instantiation), from ‘concrete’ to ‘abstract’;

Concepts shall be instantiated with individuals. Such individuals get lexical values for attributes or references for relations. Lexical values shall be classified according to some value type.

- Generalization (inverse: specialisation), from ‘specific’ to ‘generic’;

Concepts shall be specialized in other concepts; attributes shall be specialized in other attributes, and relations shall be specialized in other relations. Specialized concepts, attributes and relationships inherit all constraints and derivations of the concepts, attributes, relationships they are specialized from.

These two mechanisms give rise to two hierarchy types namely:

- a *typology*;
- a *taxonomy*.

Beside these two abstraction mechanisms the following other meta-relation should be used:

- Grouping, like for grouping all concepts, attributes, relations, etc. into one information model.

**SS-EN 17632-1:2022 (E)**

Finally, there are two more complex n-ary relations that define explicit rules that information should comply to:

- Constraints, restricting the amount of values, the values themselves or both. Concepts, value types, attributes and relations can have restrictions with respect to their source or target concepts (in case of relations) or target value type (in case of attributes); and
- Derivations, describing how new values for attributes or relations shall be inferred from existing asserted values. Constraints shall be “definitional” or represent requirements, regulations or recommendations.

## **6.2 Concrete L1: Information language bindings**

A language binding defines how the conceptual information language is mapped to the available modelling constructs provided by a given concrete information language. To keep things as simple as possible, this document uses wherever possible, the most simple and direct use of language constructs available. This means that only quantities shall be ‘objectified’ and modelled as classes, qualities and relation shall be directly modelled as RDF properties (in OWL: datatype properties respectively object properties). This way, standard language functionalities offered like inverse relations and various attribute or relation constraints types shall be often directly reused.

Below in Tables 2 and 3 the specifications are given for each modelling style per level of capability (LoC).

**Table 2 — Specified language binding for LoC-1: Terminology and definition alignment with SKOS**

<b>Meta-set/concept/relation</b>	<b>Language binding for LoC-1: Terminology and definition alignment with SKOS</b>
<b>1. Information model</b>	skos:ConceptScheme
<b>2. Information set</b>	not applicable
<b>3. Group</b>	skos:Collection
<b>4. Concept</b>	skos:Concept
<b>5. Individual</b>	in general not applicable, only enumeration items: skos:Concept
<b>6. Value type</b>	not applicable
<b>7. Value</b>	not applicable
<b>8. Attribute</b>	
8.1 Annotation	existing RDFS/SKOS/SML annotations (see 6.3.5)
8.2 Quality	skos:Concept
8.3 Quantity	skos:Concept
<b>9. Relation</b>	skos:Concept
10. Grouping	skos:inScheme, skos:member
11. Classification	skos:broader (only for enumeration type instances)
12. Generalisation	skos:broader
<b>13. Constraint</b>	not applicable
<b>14. Derivation</b>	not applicable

NOTE 1 Individuals are not modelled. Therefore, rdf:type relations are not relevant. There is one exception: the instances of “type classes” used for enumerated datatypes. Here the allowed values become SKOS concepts too, related to the broader ‘enumeration type class’.

NOTE 2 Technical representation entities involving placement in space and time are also not relevant.

**Table 3 — Specified language bindings for LoC-2&3: information exchange/sharing or integration**

Meta-set/concept/relation	Language binding for LoC-2: Information exchange/sharing with RDFS	Language binding for LoC-3a: Information integration with (RDFS+) OWL	Language binding for LoC-3b: Information integration with (RDFS+) SHACL
<b>1. Information model</b>		owl:Ontology	
<b>2. Information set</b>		owl:Ontology	
<b>3. Group</b>		rdfs:Container	
<b>4. Concept</b>	rdfs:Class	owl:Class	rdfs:Class
<b>5. Individual</b>	rdfs:Resource (implicit)	owl:NamedIndividual or anonymous individual (both implicit)	rdfs:Resource (implicit)
<b>6. Value type</b>	rdfs:Datatype, or rdfs:Class + rdf:Property for enumeration datatypes	rdfs:Datatype, or owl:Class + owl:ObjectProperty for enumeration datatypes; incl. owl:oneOf for fixed sets	rdfs:Datatype, or rdfs:Class + rdf:Property for enumeration data types; incl. sh:in for fixed sets
<b>7. Value</b>	plain or typed literal (implicit)	plain or typed literal (implicit)	plain or typed literal (implicit)
<b>8. Attribute</b>			
<b>8.1 Annotation</b>	rdf:Property	owl:AnnotationProperty	rdf:Property
<b>8.2 Quality</b>	simple via rdf:Property or complex via QualityValue range (except for enumerations)  enumerations: rdf:Property with range an instance of EnumerationType class	simple via owl:DatatypeProperty or complex via owl:ObjectProperty with QualityValue range (except for enumerations)  enumerations: owl:ObjectProperty with range an instance of EnumerationType class	simple via rdf:Property or complex via QualityValue range (except for enumerations)  enumerations: rdf:Property with range an instance of EnumerationType class

## SS-EN 17632-1:2022 (E)

Meta-set/concept/relation	Language binding for LoC-2: Information exchange/sharing with RDFS	Language binding for LoC-3a: Information integration with (RDFS+) OWL	Language binding for LoC-3b: Information integration with (RDFS+) SHACL
8.3 Quantity	simple via rdf:Property or complex via QuantityValue range	simple via owl:ObjectProperty or complex via owl:ObjectProperty with QuantityValue range	simple via rdf:Property or complex via QuantityValue range
9. Relation	simple via rdf:Property or complex via RelationReference range	simple via owl:ObjectProperty or complex via owl:ObjectProperty with RelationReference range	simple via rdf:Property or complex via RelationReference range
10. Grouping	implicit (same file) or explicit via rdfs:member		
11. Classification	rdf:type		
12. Generalisation	rdfs:subClassOf or rdfs:subPropertyOf	rdfs:subClassOf & rdfs:subPropertyOf	rdfs:subClassOf & rdfs:subPropertyOf
13. Constraint	rdfs:domain or rdfs:range	owl:Restriction rdfs:domain & rdfs:range	sh:NodeShape & sh:PropertyShape <sup>a</sup>
14. Derivation	not applicable	not applicable	sh:Rule

<sup>a</sup> When there is a domain *and* a range: mapping to a NodeShape with a PropertyShape (having a sh:targetClass in case of explicit/separate target classes). When there is a range only: mapping to a NodeShape (always having a sh:targetObjectsFrom).

NOTE 3 Since all (limited) machine-interpretable modelling really starts at LoC-2a, it will often be the starting point that might be in a later stage extended towards LoC-3 for more advanced constraint/derivation modelling.

NOTE 4 In this document, Turtle is used for encoding SML itself (Annex A) and the SML Example(Annex C).

NOTE 5 Domain and range are positioned as ‘constraint’. An RDFS reasoner infers new information based on these constraints (following the RDFS ‘entailment’). ‘Derivations’ are interpreted as user-defined inferences based on user-defined rules.

NOTE 6 The term ‘Uniform Resource Locator’ (URL) refers to the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism (e.g. its network location).

NOTE 7 The owl:AllDisjointClasses type constraints have no direct SHACL counterpart. This is modeled in SHACL as in the following example:

```
sml>AllDisjointClassesShape_1
a sh:NodeShape ;
sh:targetSubjectsOf rdf:type ;
sh:property [
  sh:path (rdf:type [sh:zeroOrMorePath rdfs:subClassOf]) ;
  sh:qualifiedValueShape [
    sh:in (
```

```

    sml:PhysicalObject
    sml:InformationObject
    sml:State
    sml:Event
    sml:Activity
  ) ;
]
sh:qualifiedMaxCount 1 ;
]
sh:property [ etc.

```

## 6.3 Modelling patterns

### 6.3.1 General

When mapping to a concrete language modelling patterns shall be applied as defined in 6.3.2 to 6.3.9.

### 6.3.2 Identification via URI strategy

A URI shall always be used to identify semantic resources following the RDF framework.

For optimal reuse and clarity the specific URI strategy should be specified as in Table 4.

**Table 4 — URI strategy**

— uri = 'http', '['s'], '://', internet domain, '/', [path], '/', reference
— reference =
— 'term', ('/'   '#'), term <sup>a</sup>
— 'def', ('/'   '#'), (conceptname   attributename   relationname   reference-individualname) <sup>b</sup>
— 'id', ('/'   '#'), individualname <sup>c</sup>
— 'doc', ('/'   '#'), documentname, [documentextension], [( '/'   '#'), fragmentidentification] <sup>d</sup>
<b>Meta-symbols</b>
= defined by (a 'production rule')
,
'...'
...
[ ... ]
( ... )
Logical 'exclusive OR'
<sup>a</sup> to denote a term for a concept, attribute or relation.
<sup>b</sup> to denote a concept, attribute, relation or reference individual.
<sup>c</sup> to denote an individual (read individual, not identifier for 'id', since terms, semantical definitions and documents also have an identifier).
<sup>d</sup> to denote a document providing information about a concept, attribute, relation or (reference) individual (optionally with fragment identification within the document).

## SS-EN 17632-1:2022 (E)

### 6.3.3 Combining language bindings

In practice, there will often be a need for a combination of language bindings. In the most extensive case, this concerns (always 1 or more):

- SKOS concept schema;
- RDFS ontology;
- OWL ontology (importing the RDFS ontology); and
- SHACL ontology (importing the RDFS ontology<sup>3</sup>).

Because the semantic intent of the SKOS concepts is fundamentally different from that of the ontological sources that describe the semantics of the information and/or serve as a basis for reasoning about information or verifying information, it is important to identify the name space URIs to distinguish the SKOS concepts from the name space URIs of the ontology sources. Also, a SKOS schema will not be imported by an RDFS/OWL/SHACL ontology. The skos:prefLabel and skos:definition will be duplicated in the RDFS ontology.

The ontological items are then possibly weakly linked to the corresponding SKOS items using the existing rdfs:seeAlso annotation.

Since RDFS, OWL and SHACL are not fundamentally different and cover complementary semantic aspects, the same name space URI can be reused between them.

The same semantic resource can therefore be simultaneously typed as an rdfs:Class, an owl:Class and a sh:NodeShape. The ontologies displayed in RDFS, OWL and SHACL are captured in their own ontology with their own 'graph URI' and can therefore also be used independently of each other and import each other.

When combining language bindings, it is therefore important to make a good distinction between 'name space URIs' (often abbreviated with prefixes) and 'graph URIs' (also called 'base URIs') that are typified as an owl:Ontology (in non-combined language bindings, name space URIs and graph URIs often coincide and a semantic software tool can generate a 'name space URI' from a 'graph URI': the so-called 'default name space').

File names are totally independent of both the name space URIs and the graph URIs.

Annex A, the 'linked data' implementation of this document, and Annex C, an application example, are designed according to these guidelines. A typical example for the combination SKOS, RDFS, OWL and SHACL from the SML code:

#### In sml-skos.ttl:

```
@prefix sml-term: <https://w3id.org/sml/term#> .
<https://w3id.org/sml/skos/term> a skos:ConceptScheme .

sml-term:PhysicalObject a skos:Concept .
```

#### In sml-rdfs.ttl:

```
@prefix sml: <https://w3id.org/sml/def#> .
@prefix sml-term: <https://w3id.org/sml/term#> .
<https://w3id.org/sml/rdfs/def> a owl:Ontology .
```

---

<sup>3</sup> Could technically also be the OWL ontology because SHACL simply ignores the OWL aspects. This is not the case the other way around: the OWL ontology should not import the SHACL graph because OWL is stricter.

```
sml:PhysicalObject a rdfs:Class ;
    rdfs:seeAlso sml-term:PhysicalObject .
```

### In sml-owl.ttl

```
@prefix sml: <https://w3id.org/sml/def#> .
<https://w3id.org/sml/owl/def> a owl:Ontology ;
    owl:imports <https://w3id.org/sml/rdfs/def> .

sml:PhysicalObject a owl:Class .
```

### In sml-shacl.ttl:

```
@prefix sml: <https://w3id.org/sml/def#> .
<https://w3id.org/sml/shacl/def> a owl:Ontology ;
    owl:imports <https://w3id.org/sml/rdfs/def> .

sml:PhysicalObject a sh:NodeShape .
```

EXAMPLE 1 <http://www.interlink.eu/roadotl/def/Road>

EXAMPLE 2 <http://www.nwb-roadnetwork.nl/#A16>

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

EXAMPLE 4 <https://w3id.org/nen2660/def#PhysicalObject>

(<https://w3id.org/nen2660/def#> is the 'name space URI')

EXAMPLE 5 <https://w3id.org/nen2660/shacl/def> (example of a 'graph URI')

NOTE 1 The choice between "/" or "#" depends on the expected size of the information models/information sets. If big: use "/", after '#' there is no more server-control, more background information in [4].

NOTE 2 If there is a need for versioning in the URI, then the following approach is specified (versionless and versioned variants where the versionless always refers to the latest version):

uri = 'http', ['s'], '://', internet domain, '/', [path], '/', [version], '/', reference

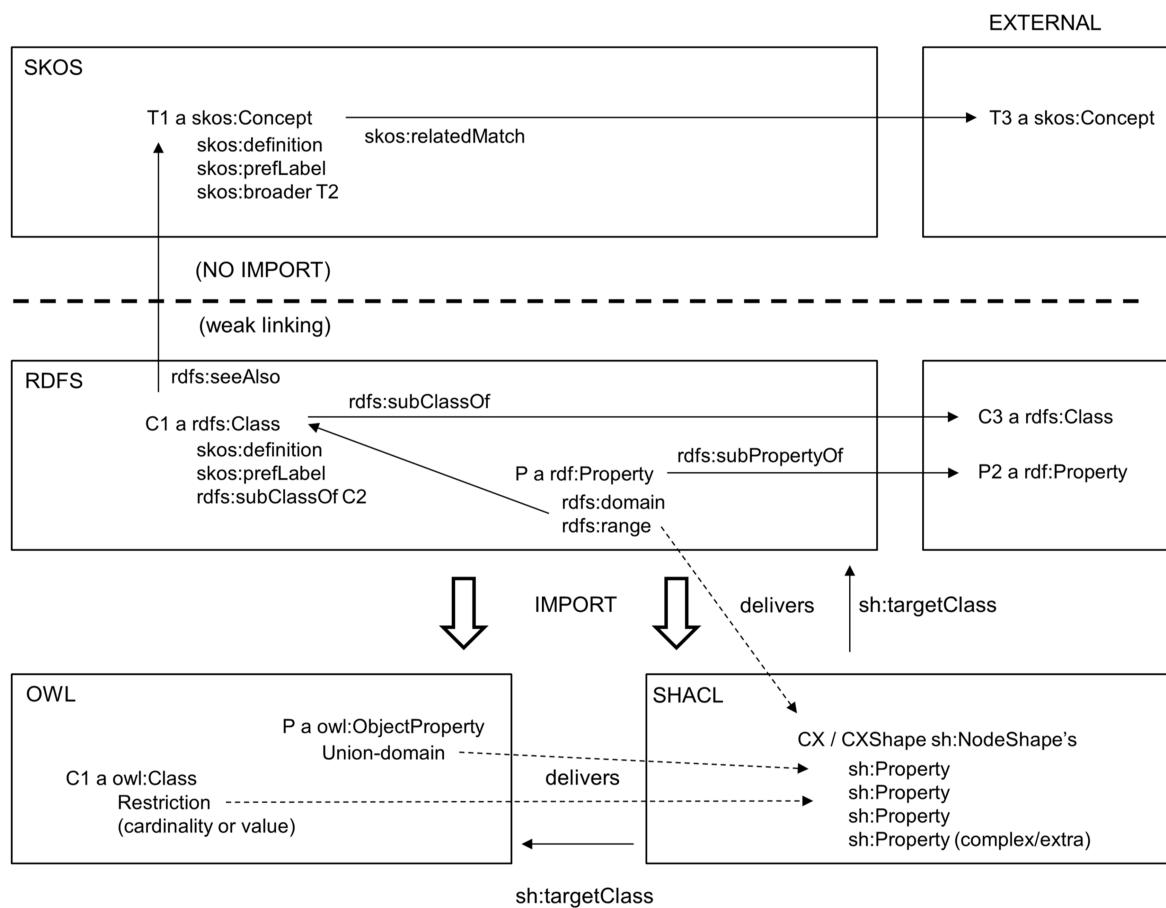
NOTE 3 The graph URIs are preferably not only for identification but also for localization (making information models, information sets and their elements 'dereferenceable').

NOTE 4 Do not use file extensions in names of information models and information sets. Use standard 'server negotiation' to select the correct format (including human-readable texts in i.e. .html files).

NOTE 5 If necessary, use internet services that offer stable URIs such as w3id.org or purl.org. In that case, a path is often needed since generic fragments like 'def' have been claimed.

Content-wise, a typical combination in depicted in Figure 3.

## SS-EN 17632-1:2022 (E)

**Figure 3 — Multi-RDF-language modelling**

Note that some of the SHACL shapes are generated from the RDFS/OWL restrictions so that they can be used for information verification instead of information inference.

Having Figure 3 in mind an end-user ontology shall be of one of the following options:

- A LoC-1 SKOS scheme only;
- A LoC-2 RDFS ontology only;
- A LoC-2 RDFS ontology connected via **rdfs:seeAlso** links to a SKOS scheme providing also the **skos:prefLabel**, **skos:definition**, **skos:broader** weak taxonomy matching the RDFS **rdfs:subClassOf** strong taxonomy and possibly more SKOS-level information;
- A LoC-3 OWL ontology only, including all RDFS statements;
- A LoC-3 OWL ontology where the classes are also typed as RDFS classes available in a separate ontology. In the OWL ontology only the OWL specific statements are modelled like the distinction in **owl:DatatypeProperty** and **owl:ObjectProperty** and more advanced variants of **rdfs:domain** and **rdfs:range** involving **owl:unionOf**;
- A LoC-3 SHACL ontology only, including all RDFS statements;

- A LoC-3 SHACL ontology where the classes are also typed as RDFS classes (or explicitly related to via target classes) in a separate ontology. In the SHACL ontology only the SHACL shapes are modelled partly mimicing those in RDFS (rdfs:domain/rdfs:range). SKOS separate or not; and
- A LoC-3 combination of RDFS, OWL and SHACL where in SHACL now also the OWL restrictions are mimicked. SKOS separate or not.

For the implementation (Annex A) and typical example (Annex C) of this document the last, full option is used showing a separation of concern where the different RDF languages all contribute to different modelling aspects.

### 6.3.4 Naming

There are two primary ways to uniquely name things in information sets or information models (last fragment of URI):

- by meaningful names (for humans); or
- by semantic-free, opaque codes (like a generated UUID code).

A third option would be any combination of these two ways (“Road-A237-12”).

All approaches have their advantages and disadvantages and are allowed by this document. When developing and learning ontologies human-friendly names help while in an operational phase supported by semantic software tools hiding the details, codes are typically more stable (they stay stable when terms change) and language-neutral. Codes also make it easier to deal with homonyms (terms having multiple meanings).

Where codes are used, human-friendly preferred labels in the context of one ontology shall be used too, since alternative labels are not unique.

When using names in information models:

- Meaningful concept or value type names always start with a capital (upper case) letter (“CamelCase”);
- Meaningful attribute or relation names always start with a non-capital (lower case) letter (“camelCase”); and
- Meaningful names are singular.

Example names in information sets are flexible, e.g.:

- Specific names like “A16” for a Dutch road;
- Numbered individuals for a class like “Road\_16”;
- Semantic-free assigned GUID’s like “233hhjj334--!!99”;
- Semantic-free generated UUID’s like “550e8400-e29b-41d4-a716-446655440000” (typically a bit long but with the advantage of auto-generation); or
- Any combination of the above options.

**SS-EN 17632-1:2022 (E)****6.3.5 Annotation**

Internal textual definitions shall reuse the simple knowledge organization system (SKOS):

- <http://www.w3.org/2004/02/skos/core#definition> (rdf:type rdf:Property);
- There shall be maximum one definition per language code.

The use of skos:definition provides more specific meaning than the generic rdfs:comment.

In addition, rdfs:isDefinedBy is used to point to an external resource defining the subject resource. This property may be used to indicate an RDF vocabulary in which a subject resource is defined. It points to the authoritative information about the resource (which are not necessarily RDF, often html, and in some cases native formats like PDF).

Codes (independent of any language) shall be defined again using SKOS:

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

Each of the used codes/notations shall uniquely identify the things having the code/notation.

Overview of selected annotation types specified by this document is given in Table 5.

**Table 5 — Specified language binding for annotations**

Annotation	Language binding
primary label	skos:prefLabel
synonym	skos:altLabel
code	skos:notation
abbreviation	sml:abbreviation (rdfs:subPropertyOf of skos:altLabel)
internal definition	skos:definition (max 1 per language)
external definition	rdfs:isDefinedBy
external description	rdfs:seeAlso (used for link to SKOS scheme)
scope	skos:scopeNote
comment	rdfs:comment
remark by author	skos:editorialNote
example application	skos:example
source of information	dcterms:source

Of course, other annotation properties can be added like i.e. 'dcterms:created'.

For a common look and feel in viewers/editors this document specifies that labels:

- start with a capital (upper case) letter for concepts and datatypes;
- start with a non-capital (lower case) letter for attributes and relations;
- use no other capital letters (unless for acronyms);

- use spaces to separate words;
- do not use dots in the end;
- always use language tags (possibly including regions) according to [3]. Examples: "Bridge"@en, "Pechstrook"@nl-BE; and
- when using user-friendly names copy them as preferred label too, applying the rules above (if the class name is "InnerDoor" make the SKOS preferred label also "Inner door").

### 6.3.6 Types

Two kinds of types are distinguished: enumeration types and concept types.

Enumeration types are specified as a concept being an instance of an EnumerationType so that allowed enumeration items can become individuals with multi-lingual labels and other metadata like descriptions, definitions, etc. Open versus fixed enumeration datatypes are distinguished. The fixed ones are specified using the OWL (default-OWA) 'owl:oneOf' construct; having a SHACL (default-CWA) 'sh:in' counterpart.

The name of the enumeration datatype should be a capital-starting variant of the relationship having this datatype as range. For clarity is advised to extend both the property and the underlying type with a "Type-like" fragment. The EnumerationType itself is modelled as an isSubtypeOf rdfs:Container to be able to specify an order for the enumeration items.

#### EXAMPLE (in OWL)

```

ex:loadLevelType
  a owl:ObjectProperty ;
  rdfs:domain ex:Vehicle ;
  rdfs:range ex:LoadLevelType ;

ex:LoadLevelType
  a owl:Class ;
  a sml:EnumerationType ;
  owl:oneOf (
    ex:Light
    ex:Normal
    ex:Heavy
  ) ;
  rdf:_1 ex:Light ;
  rdf:_2 ex:Normal ;
  rdf:_3 ex:Heavy .

ex:Light a ex:LoadLevelType .
ex:Normal a ex:LoadLevelType .
ex:Heavy a ex:LoadLevelType .

```

This can also be represented in RDFS, loosing the 'one of' constraint:

```

ex:loadLevelType
  a rdf:Property ;
  rdfs:domain ex:Vehicle ;

```

**SS-EN 17632-1:2022 (E)**

```
rdfs:range ex:LoadLevelType .

ex:LoadLevelType
  a rdfs:Class ;
  a sml:EnumerationType ;
  rdf:_1 ex:Light ;
  rdf:_2 ex:Normal ;
  rdf:_3 ex:Heavy .

ex:Light a ex:LoadLevelType .
ex:Normal a ex:LoadLevelType .
ex:Heavy a ex:LoadLevelType .
```

NOTE 1 Enumeration items can be a member of multiple enumeration datatypes.

Besides, enumeration types, also concept types (ConceptType) can be used as basic building block for meta-'information models' where instances are types of things, not individuals. Contrary to enumeration types, concept types can have valued attributes and relations beyond annotations and can be further instantiated towards individuals.

**EXAMPLE** (language-independent)

```
Concept CarType
  isOfType ConceptType
Concept LexusGS450h
  isOfType CarType
  systemPower 345HP
Individual MyCar
  isOfType LexusGS450h
  registration 4-KJT-99
```

NOTE 2 Concept types are the basic building block of meta-'information models' (also referred to as 'knowledge models'). EN ISO 12006-3 uses the term 'xtdSubject'.

### 6.3.7 Decomposition (instance level)

This document specifies a non-transitive hasPart relation that is not necessarily inverse functional. The transitive variant shall be derived when needed. A meronomy (typical decomposition) shall be defined by constraining this hasPart relation.

NOTE This abstraction mechanism is special since it is not directly supported by the languages used as already apparent from the earlier language bindings.

### 6.3.8 Quantity kinds & units

This document specifies the QUDT [2] ontology published by qudt.org for quantity kinds and units.

NOTE QUDT is fully aligned with the EN ISO 80000 series of standards [10] (system, names, definitions, symbols, etc.).

### 6.3.9 Complex properties

There are many ways to model properties that are attributes (qualities or quantities) and relations. All have advantages and disadvantages. The most simple and direct modelling of an attribute in e.g. in OWL by an owl:DatatypeProperty. This way it is however hard to model compound values (like a nominal value and a unit) or metadata for values.

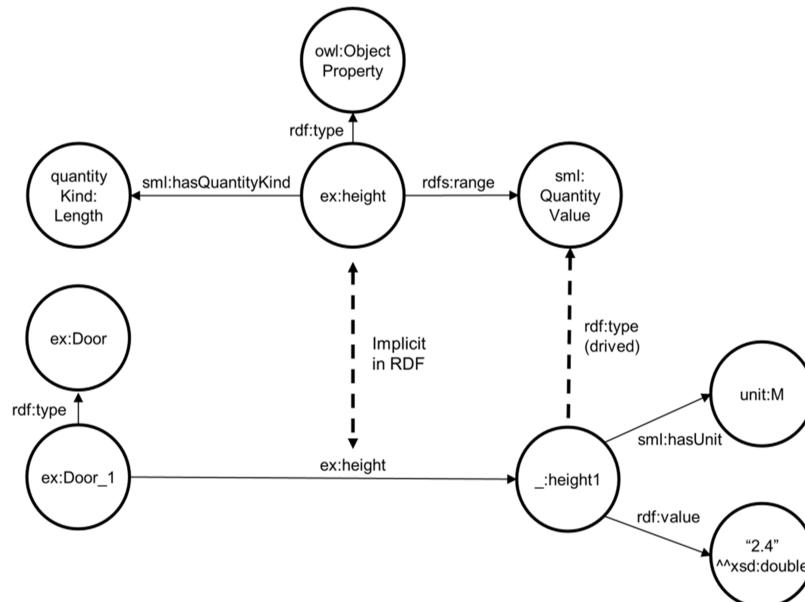
Therefore this document specifies modelling patterns for both complex values for attributes (qualities and quantities) and relations. These properties are defined as relations towards these complex values. This pattern is fully in line with 'best practice pattern no. 1' in [6]. Depending on the need, a direct/simple or indirect/complex variant can be chosen; possibly even mixed.

Quantities in the complex variant shall be modeled as relations (as in OWL with an owl:ObjectProperty) with the range (rdfs:range) being the concept 'QuantityValue'. A simple value itself (via rdf:value with range xsd:decimal, xsd:float or xsd:double) and other attributes as unit (via sml:hasUnit) etc. can be attached to this complex quantity value. That simple value is optional and only used as a 'nominal value' where the other components are secondary in the composition (like a unit) or are all metadata about this value.

A quantity type can be attached to a quantity definition (via sml:hasQuantityKind) so actually to the relation itself using 'punning'.

The patterns can be applied recursively, for example, to model a compound quantity of subquantities themselves having a value and a unit.

A typical example of a (non-recursive) complex quantity is given in Figure 4.



**Figure 4 — Example complex quantity modelling**

**NOTE 1** Because complex quality/quantity/relation values do not have to be reused, they can remain anonymous at the instance level (a so-called "blank node"). Also, typing the instance as a sml:QualityValue, sml:QuantityValue or sml:RelationReference is not necessary. By giving each attribute such range on class-level, this typing of the instances can be inferred.

**NOTE 2** The same approach as for quantities is used for complex qualities and relations. The only difference is that the range of the optionally used rdf:value is different: not a Decimal/Float (xsd:decimal/xsd:float/xsd:double) but a String (xsd:string) or else a URI (xsd:anyURI).

**SS-EN 17632-1:2022 (E)****7 M1: Information model****7.1 Top level information model**

This document specifies the following top level information model concepts and their relations and (hasPart) constraints:

- TopConcept
  - AbstractConcept
    - Type
      - EnumerationType
      - ConceptType
    - Objectification
      - QualityValue
      - QuantityValue
      - RelationReference
  - ConcreteConcept
    - Entity
      - Object
        - PhysicalObject
        - InformationObject
          - Representation
            - GeometricEntity
            - TemporalEntity
      - Activity
      - State
      - Event

A distinction is made in two kinds of entities: objects (subdivided into physical objects and information objects) that *exist* and activities that *happen*. The language introduced earlier does not yet introduce ‘time aspects’ so this has to be taken care of by the actual conceptual modelling itself, here by the introduction of ‘dynamic’ concepts beyond those ‘static’ objects and activities being states and events. When time is not relevant in case of timeless static aspects or just a ‘snapshot’ of entities in time, these states and events can be ignored (kept implicit).

There are two main relationships between objects and activities: physical objects perform activities and activities transform objects (physical objects or information objects).

Entities, so both objects and activities, live in space(time) and shall have a relevant positioning in space and time (via a spatial entity respectively a temporal entity being special information objects).

A spatial entity shall be used for two purposes: defining the topological interior and boundary of a physical object.

**EXAMPLE** Specifying the boundary of a functional transport network a 3D corridor network is defined. This shall be accomplished by aggregating the boundaries of the edges within the network which themselves are topologically connected via nodes.

This document only specifies the placeholder constructs to define geometry (location/explicit shape) in space. The required details have to come from existing spatial representation standards such as: ISO 16739-1, OGC's GML, CityGML, GeoSPARQL, W3C's wgs84\_pos, IETF's GeoJSON, CityJSON, etc. See for more information on (web-based) spatial information standards [12]. The semantics of the general relation towards temporal entities depends on the domain: events have a time instant, entities (objects and activities) a time interval.

Finally, the top level information model specifies relations, restricted to their source and target concepts. They are specified in Table 6, their inverses in Table 7.

**Table 6 — Specified top level relations**

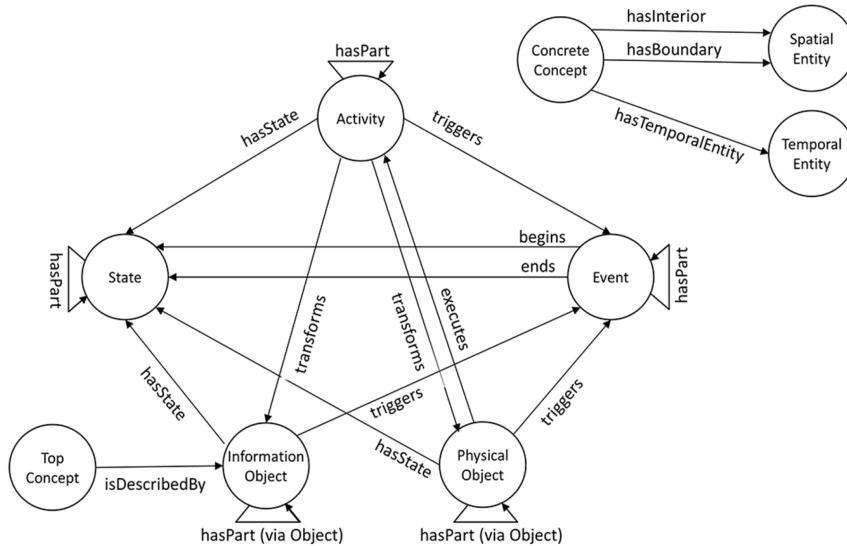
Domain	Relation	Range
TopConcept	isDescribedBy	InformationObject
ConcreteConcept	hasPart	ConcreteObject
Entity	hasState	State
Entity	triggers	Event
PhysicalObject	executes	Activity
ConcreteConcept	hasTemporalEntity	TemporalEntity
ConcreteConcept	hasInterior	SpatialEntity
ConcreteConcept	hasBoundary	SpatialEntity
Activity	transforms	Object
Event	begins	State
Event	ends	State

**Table 7 — Specified top level inverse relations**

Domain	Relation	Range
InformationObject	describes	TopConcept
ConcreteConcept	isPartOf	ConcreteConcept
State	isStateOf	Entity
Event	isTriggeredBy	Entity
Activity	isPerformedBy	PhysicalObject
TemporalEntity	isTemporalEntityFor	ConcreteConcept
SpatialEntity	isInteriorOf	ConcreteConcept
SpatialEntity	isBoundaryOf	ConcreteConcept
Object	isTransformedBy	Activity
State	isBegunBy	Event
State	isEndedBy	Event

**SS-EN 17632-1:2022 (E)**

When graphically combining the key top level information model concepts and relations, Figure 5 is obtained (inverses and some generic concepts left out).



**Figure 5 — Predefined concepts and relations combined**

## 7.2 Systems engineering extension

All earlier distinguished archetypical concepts (physical object, information object, activity, state and event) can in principle be further subclassed along two extra dimensions. According to this document (only) an entity (object or activity) shall be further subclassed:

- Functional entity versus technical entity;
- Planned entity versus realized entity.

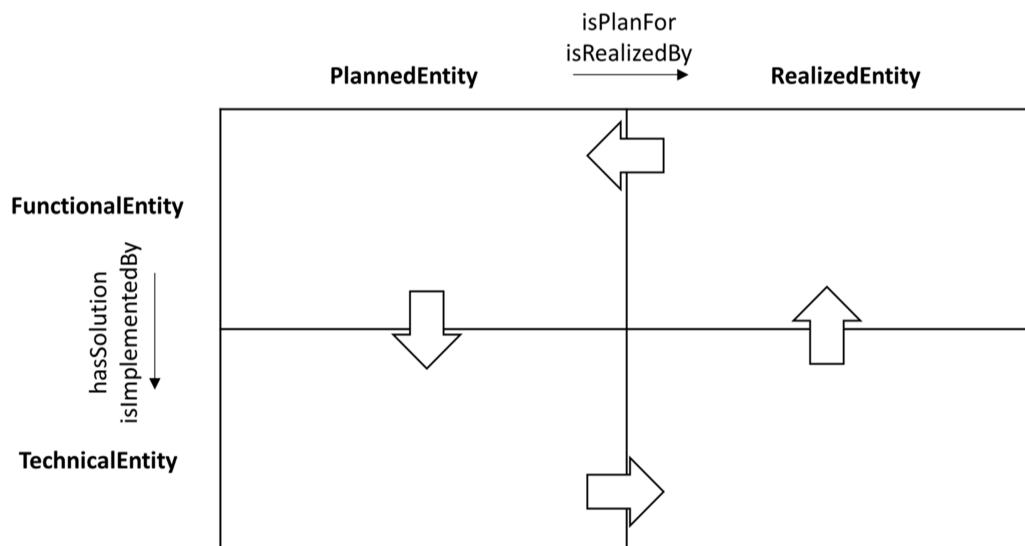
**NOTE 1** If an individual is classified as a kind (of object) (here referred to as a ‘technical entity’) this classification will normally persist for its whole life. For example, if something is a book, it will normally always be a book. It is about what something “is”. If an individual is classified as a functional entity it is often related to a certain period in time. If someone is a father he was that normally from a certain starting point in time. So, someone “is” not a father, but plays the role of a father. He might also play that role multiple times in parallel or subsequently in time.

**NOTE 2** This dimension is completely orthogonal with “typical decomposition” (aka ‘decomposition on class level): the distinction between technical and functional can be made on any scale level: networks, complexes, assets, elements, components and materials.

**NOTE 3** Functional objects can be decomposed by a client into sub-functional objects where the leaves of the decomposition tree are implemented in technical objects by contractors/suppliers providing make/buy solutions that themselves decompose further in sub-technical objects. Clearly, this process can be recursive involving multiple “Functional-Technical” jumps by subsequent parties in the supply chain.

When the two dimensions are combined (Figure 6), four extra subclasses of physical object are implicitly ‘generated’. The arrows in the quadrants indicate a typical process flow in practice: an asset is functionally programmed, technically designed and technically built in such a way that it functionally

performs in reality the way it was specified. At later points in time the asset can be re-programmed, re-designed or re-built (at its end of life, ultimately demolished and recycled).



**Figure 6 — Four quadrants relevant for systems engineering**

In particular, a functional physical object, as a subtype of both a physical object and a functional entity, executes a functional activity (often referred to as a ‘function’). Like activity, this functional activity always takes the form of a verb (not a noun) like “connecting”, “pumping”, “purchasing”, etc.

SML constructs summarized:

- sml:FunctionalEntity;
- sml:TechnicalEntity;
- sml:PlannedEntity;
- sml:RealizedEntity;
- sml:hasSolution (from sml:FunctionalEntity to sml:TechnicalEntity);
- sml:isImplementedBy (from sml:FunctionalEntity to sml:TechnicalEntity);
- sml:isPlanFor (from sml:PlannedEntity to sml:RealizedEntity);
- sml:isRealizedBy (from sml:PlannedEntity to sml:RealizedEntity).

The two subproperties indicate a chosen solution or else a chosen realization from sets of potential (alternative) solutions/realizations.

The instances, typified as 1 or more of these options, all receive their own identifier so that they can also be related to each other.

NOTE 4 Alternatively, it is possible to work with ‘main states’ of the same object, but this is difficult to achieve in practice because often not one source holder can be designated over the entire life cycle.

## SS-EN 17632-1:2022 (E)

NOTE 5 For each quadrant (per 'main state' modeled with the 2 dimensions) you can optionally work with different (sub)states of an asset. This makes it possible to make several rounds in the quadrants so that not only the first creation of assets is supported, but also maintenance, renovation and repurposing (involving more and more mutations in more quadrants).

## 8 Implementing SML in code

The above requirements are implemented in the combination of the four languages (SKOS, RDFS, OWL and SHACL) in the normative Annex A. The four specified concrete language subsets are specified in the normative Annex B.

NOTE 1 An informative example is elaborated in the same way in Annex C.

NOTE 2 The RDF serialization used for this implementation is Turtle (for all languages). Other RDF serializations can be easily derived by existing translators without loss of information.

## 9 Conformance

### 9.1 General

This clause defines two levels of conformance for information sets.

The actual conformance required in practice can be specified in exchange information requirements (EIR).

### 9.2 Conformance on language level

An information set conforms on language level if, as prescribed by this document, it is:

1. Expressed according to a standard RDF serialization (XML, Turtle or JSON-LD), and/or directly accessible via the standard SPARQL 1.1 query language.
2. In case of explicit information models, making the information sets 'semantic', the standard linked information languages (like RDF, SKOS, RDFS, OWL and/or SHACL) and the modelling patterns as prescribed by this document are used. As a prerequisite, these information models are also expressed according to a standard RDF serialization as used for the information sets.

NOTE Equivalent RDF formats include XML, Turtle and JSON-LD.

Information models, reflect asset information from any perspective, including:

- Software-specific semantics;
- Person-specific semantics;
- Project-specific semantics;
- Department-specific semantics;
- Organization-specific semantics;
- Country-specific semantics; and
- Fully generic, International semantics.

Semantics can still be software-specific but at least this semantics is explicitly modelled and therefore more easily transformable towards any other semantics. Therefore, information is both syntactically and semantically 'liberated from the software'.

### **9.3 Conformance on semantic level**

An information set conforms on semantic level if it reuses the generic top level information model (expressed in one of the standard RDF serialization and using one of the standard 'linked data' languages) as specified by this document.

This conformance on semantic level only makes sense on top of the conformance on language level.

**SS-EN 17632-1:2022 (E)****Annex A**  
(normative)**SML implementation in 'linked data'****A.1 Introduction**

This annex describes the SML implementation code in the combined 'linked data' (RDF-based) languages (SKOS, RDFS, OWL and SHACL), in the RDF serialization Turtle. It covers both the in this document defined conceptual information language and generic top level information model.

The basic language vocabularies for XSD, RDF, RDFS and OWL are expected to be 'automatic' imports i.e. there is no explicit owl:imports clause needed for them. Note that SKOS and SHACL are not in that list; most semantic software tools need an explicit import or they ask users to add it when creating a SKOS respectively a SHACL file. They are here always imported. Others are imported when needed.

Prefixes will be specified for:

- the 'name space URI' (SML-term or SML);
- all automatic imports (XSD, RDF, RDFS and OWL);
- all non-automatic imports needed for SML (skos, sh, qudt, unit, quantitykind and time); and
- SML-term in the SML RDFS part (note that all non-SKOS resources have the same 'name space URI' hence prefix being SML).

Often these imports/prefixes are too much (more than needed) but this is accepted for uniformity.

- The used 'name space URIs' are:
  - <https://w3id.org/sml/term> for SKOS (prefix: sml-term); and
  - <https://w3id.org/sml/def> for RDFS/OWL/SHACL (prefix: sml).
- The used 'graph URIs' are:
  - <https://w3id.org/def/sml/skos/term> for SKOS (file: sml-skos.ttl);
  - <https://w3id.org/def/sml/rdfs/def> for RDFS (file: sml-rdfs.ttl);
  - <https://w3id.org/def/sml/owl/def> for OWL (file: sml-owl.ttl); and
  - <https://w3id.org/def/sml/shacl/def> for SKOS (file: sml-shacl.ttl).

This approach can also be applied for end-user ontologies making use of SML (like the typical example in Annex C).

**A.2 SKOS part**

```
@prefix sml-term: <https://w3id.org/sml/term#> .
@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 sh: <http://www.w3.org/ns/shacl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix unit: <http://qudt.org/vocab/unit> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/sml/skos/term>
  a skos:ConceptScheme ;
  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/2004/0/skos/core> ;
  owl:imports <http://www.w3.org/2006/time> ;
  owl:imports <http://www.w3.org/ns/shacl> ;
  skos:hasTopConcept sml-term:TopConcept ;
.

sml-term:AbstractConcept
  a skos:Concept ;
  skos:broader sml-term:TopConcept ;
  skos:definition "Concept that forms a manifestation and demarcation in an abstract space. An abstract concept has no life cycle"@en ;
  skos:prefLabel "Abstract concept"@en ;
.

sml-term:Activity
  a skos:Concept ;
  skos:broader sml-term:Entity ;
  skos:definition "Entity that takes place or can take place in a concrete space-time. An activity transforms objects, and is executed by an object"@en ;
  skos:prefLabel "Activity"@en ;
.

sml-term:ConceptType
  a skos:Concept ;
  skos:broader sml-term:Type ;
  skos:definition "A meta-concept having concepts as instances that can have attributes or relations"@en ;
  skos:prefLabel "Concept type"@en ;
.

sml-term:ConcreteConcept
  a skos:Concept ;
  skos:broader sml-term:TopConcept ;
  skos:definition "Concept that forms a manifestation and a demarcation in a concrete space-time. A concrete concept has a life cycle"@en ;
  skos:prefLabel "Concrete concept"@en ;
.

sml-term:Entity
  a skos:Concept ;
  skos:broader sml-term:ConcreteConcept ;
  skos:definition "Concept that forms a manifestation and a demarcation in a concrete space-time, and which has a certain state at any moment in time. An entity has a unique identity that remains constant throughout its life cycle. The life cycle of an entity is made up of the sequence of

```

**SS-EN 17632-1:2022 (E)**

states of that entity, which follow each other in time. An entity is an object or an activity. An object exists, an activity takes place"@en ;  
 skos:prefLabel "Entity"@en ;  
 .

sml-term:EnumerationType  
 a skos:Concept ;  
 skos:broader sml-term:Type ;  
 skos:definition "A meta-concept having concepts as instances having no further attributes or relations (annotations only)"@en ;  
 skos:prefLabel "Enumeration type"@en ;  
 .

sml-term:Event  
 a skos:Concept ;  
 skos:broader sml-term:ConcreteConcept ;  
 skos:definition "Transition between two successive states of an entity (object or activity). An event is triggered in a state"@en ;  
 skos:prefLabel "Event"@en ;  
 .

sml-term:FunctionalEntity  
 a skos:Concept ;  
 skos:broader sml-term:Entity ;  
 skos:definition "An entity involving the external behavior where the output contributes to stakeholder objectives implemented/played by one or more technical entities"@en ;  
 skos:prefLabel "Functional entity"@en ;  
 .

sml-term:GeometricEntity  
 a skos:Concept ;  
 skos:broader sml-term:Representation ;  
 skos:definition "Named concept, which forms an actual or virtual demarcation in a concrete (physical, three-dimensional) space, which we experience in reality"@en ;  
 skos:prefLabel "Geometric entity"@en ;  
 .

sml-term:InformationObject  
 a skos:Concept ;  
 skos:broader sml-term:Object ;  
 skos:definition "Object which describes a thing in reality"@en ;  
 skos:prefLabel "Information object"@en ;  
 .

sml-term:Object  
 a skos:Concept ;  
 skos:broader sml-term:Entity ;  
 skos:definition "Entity that exists or can exist within a concrete space-time. An object executes an activity, and is transformed by an activity"@en ;  
 skos:prefLabel "Object"@en ;  
 .

sml-term:Objectification  
 a skos:Concept ;  
 skos:broader sml-term:AbstractConcept ;  
 skos:definition "An attribute (quality or quantity) or relation represented as an individual defined by a concept"@en ;  
 skos:prefLabel "Objectification"@en ;  
 .

```
sml-term:PhysicalObject
  a skos:Concept ;
  skos:broader sml-term:Object ;
  skos:definition "Object that exists or may exist within physical 4D space-time. A physical object forms a manifestation and a demarcation of matter and/or energy, and is (in)directly perceptible by the senses"@en ;
  skos:prefLabel "Physical object"@en ;
.

sml-term:PlannedEntity
  a skos:Concept ;
  skos:broader sml-term:Entity ;
  skos:definition "An entity that does not yet exist in physical reality, but which exists in mental or conceptual reality"@en ;
  skos:prefLabel "Planned entity"@en ;
.

sml-term:QualityValue
  a skos:Concept ;
  skos:broader sml-term:Objectification ;
  skos:definition "The objectification of a value of a quality having a complex value like a simple value sec combined with other metadata or just a combination of simple values"@en ;
  skos:prefLabel "Quality value"@en ;
.

sml-term:QuantityKind
  a skos:Concept ;
  skos:definition "Any observable property that can be measured and quantified numerically"@en ;
  skos:prefLabel "Quantity kind"@en ;
.

sml-term:QuantityValue
  a skos:Concept ;
  skos:broader sml-term:Objectification ;
  skos:definition "The objectification of a value of a quantity (typically involving a quantity kind and a unit) having a complex value like a simple value sec combined with other metadata or just a combination of simple values"@en ;
  skos:prefLabel "Quantity value"@en ;
.

sml-term:RealizedEntity
  a skos:Concept ;
  skos:broader sml-term:Entity ;
  skos:definition "An entity that exists or has existed in the physical reality"@en ;
  skos:prefLabel "Realized entity"@en ;
.

sml-term:RelationReference
  a skos:Concept ;
  skos:broader sml-term:Objectification ;
  skos:definition "The objectification of a reference for a relation having a complex value like a simple reference sec combined with other metadata or just a combination of simple references (n-ary relation)"@en ;
  skos:prefLabel "Relation reference"@en ;
.

sml-term:Representation
```

**SS-EN 17632-1:2022 (E)**

```

a skos:Concept ;
skos:broader sml-term:InformationObject ;
skos:definition "That which represents something else"@en ;
skos:prefLabel "Representation"@en ;
.

sml-term:State
a skos:Concept ;
skos:broader sml-term:ConcreteConcept ;
skos:definition "Temporal part of an entity during a period between two events. A state is characterized by the properties and relations of the entity"@en ;
skos:prefLabel "State"@en ;
.

sml-term:TechnicalEntity
a skos:Concept ;
skos:broader sml-term:Entity ;
skos:definition "An entity concerned with technical properties and relations that implements or plays functional entities"@en ;
skos:prefLabel "Technical entity"@en ;
.

sml-term:TemporalEntity
a skos:Concept ;
skos:broader sml-term:Representation ;
skos:definition "A temporal interval or instant"@en ;
skos:prefLabel "Temporal entity"@en ;
.

sml-term:TopConcept
a skos:Concept ;
skos:definition "The most generic concept"@en ;
skos:prefLabel "Top concept"@en ;
.

sml-term:Type
a skos:Concept ;
skos:broader sml-term:AbstractConcept ;
skos:definition "A meta-concept which instances are concepts (not individuals)"@en ;
skos:prefLabel "Type"@en ;
.

sml-term:Unit
a skos:Concept ;
skos:definition "A particular quantity value that has been chosen as a scale for measuring other quantities the same kind"@en ;
skos:prefLabel "Unit"@en ;
.

sml-term:abbreviation
a skos:Concept ;
skos:broader skos:altLabel ;
skos:definition "Acronym or initial word as special case of a name"@en ;
skos:prefLabel "abbreviation"@en ;
.

sml-term:begins
a skos:Concept ;
skos:definition "The state that is started by an event"@en ;
skos:prefLabel "begins"@en ;

```

```
. sml-term:ends
  a skos:Concept ;
  skos:definition "The state that is ended by an event"@en ;
  skos:prefLabel "ends"@en ;

. sml-term:executes
  a skos:Concept ;
  skos:definition "The activity executed by a physical object"@en ;
  skos:prefLabel "executes"@en ;

. sml-term:hasBoundary
  a skos:Concept ;
  skos:definition "The geometric entity representing the boundary of a
concrete concept"@en ;
  skos:prefLabel "has boundary"@en ;

. sml-term:hasFunctionalPart
  a skos:Concept ;
  skos:broader sml-term:hasPart ;
  skos:definition "The hasPart relation of a functional nature"@en ;
  skos:prefLabel "has functional part"@en ;

. sml-term:hasInterior
  a skos:Concept ;
  skos:definition "The geometric entity representing the interior of a
concrete concept"@en ;
  skos:prefLabel "has interior"@en ;

. sml-term:hasPart
  a skos:Concept ;
  skos:definition "A decomposition (hasPart) relation between concrete
concepts"@en ;
  skos:prefLabel "has part"@en ;

. sml-term:hasQuantityKind
  a skos:Concept ;
  skos:definition "The possession of a quantity kind"@en ;
  skos:prefLabel "has quantity kind"@en ;

. sml-term:hasSolution
  a skos:Concept ;
  skos:definition "A technical entity that is a potential solution for a
functional entity"@en ;
  skos:prefLabel "has solution"@en ;

. sml-term:hasState
  a skos:Concept ;
  skos:definition "The state of an entity in a certain point or period of
time"@en ;
  skos:prefLabel "has state"@en ;

. sml-term:hasTechnicalPart
  a skos:Concept ;
  skos:broader sml-term:hasPart ;
```

**SS-EN 17632-1:2022 (E)**

```

skos:definition "A hasPart relation of a technical nature"@en ;
skos:prefLabel "has technical part"@en ;

.

sml-term:hasTemporalEntity
a skos:Concept ;
skos:definition "The abstract temporal entity representing a concrete
concept in time"@en ;
skos:prefLabel "has temporal entity"@en ;

.

sml-term:hasUnit
a skos:Concept ;
skos:definition "The possession of a unit"@en ;
skos:prefLabel "has unit"@en ;

.

sml-term:isDescribedBy
a skos:Concept ;
skos:definition "The information object that describes something"@en ;
skos:prefLabel "is described by"@en ;

.

sml-term:isImplementedBy
a skos:Concept ;
skos:broader sml-term:hasSolution ;
skos:definition "The technical entity that implements a functional
entity"@en ;
skos:prefLabel "is implemented by"@en ;

.

sml-term:isPlanFor
a skos:Concept ;
skos:definition "A potentially realized entity according to this
planned entity"@en ;
skos:prefLabel "is plan for"@en ;

.

sml-term:isRealizedBy
a skos:Concept ;
skos:broader sml-term:isPlanFor ;
skos:definition "The realized entity that realizes a planned entity"@en ;
skos:prefLabel "is realized by"@en ;

.

sml-term:transforms
a skos:Concept ;
skos:definition "The object transformed by an activity"@en ;
skos:prefLabel "transforms"@en ;

.

sml-term:triggers
a skos:Concept ;
skos:definition "The event triggered by an entity"@en ;
skos:prefLabel "triggers"@en ;

```

**A.3 RDFS part**

**NOTE** The typing to RDFS classes is not really needed since it can be inferred from the typing to OWL classes. For clarity and potential separate use they are made explicit.

```

@prefix sml: <https://w3id.org/sml/def#> .
@prefix sml-term: <https://w3id.org/sml/term#> .
@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 sh: <http://www.w3.org/ns/shacl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

qudt:QuantityKind
a sml:EnumerationType ;
rdfs:seeAlso sml-term:QuantityKind ;
skos:definition "Any observable property that can be measured and quantified numerically"@en ;
skos:prefLabel "Quantity kind"@en ;
.

qudt:Unit
a sml:EnumerationType ;
rdfs:seeAlso sml-term:Unit ;
skos:definition "A particular quantity value that has been chosen as a scale for measuring other quantities the same kind"@en ;
skos:prefLabel "Unit"@en ;
.

time:TemporalEntity
rdfs:seeAlso sml-term:TemporalEntity ;
rdfs:subClassOf sml:Representation ;
skos:definition "A temporal interval or instant"@en ;
skos:prefLabel "Temporal entity"@en ;
.

<https://w3id.org/sml/rdfs/def>
a 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> ;
owl:imports <http://www.w3.org/2006/time> ;
owl:imports <http://www.w3.org/ns/shacl> ;
.

sml:AbstractConcept
a rdfs:Class ;
rdfs:seeAlso sml-term:AbstractConcept ;
rdfs:subClassOf sml:TopConcept ;
skos:definition "Concept that forms a manifestation and demarcation in an abstract space. An abstract concept has no life cycle"@en ;
skos:prefLabel "Abstract concept"@en ;
.

sml:Activity
a rdfs:Class ;
rdfs:seeAlso sml-term:Activity ;
rdfs:subClassOf sml:Entity ;

```

**SS-EN 17632-1:2022 (E)**

```

skos:definition "Entity that takes place or can take place in a
concrete space-time. An activity transforms objects, and is executed by
an object"@en ;
  skos:prefLabel "Activity"@en ;

.

sml:ConceptType
  a rdfs:Class ;
  rdfs:seeAlso sml-term:ConceptType ;
  rdfs:subClassOf sml:Type ;
  skos:definition "A meta-concept having concepts as instances that can
have attributes or relations"@en ;
  skos:prefLabel "Concept type"@en ;

.

sml:ConcreteConcept
  a rdfs:Class ;
  rdfs:seeAlso sml-term:ConcreteConcept ;
  rdfs:subClassOf sml:TopConcept ;
  skos:definition "Concept that forms a manifestation and a demarcation
in a concrete space-time. A concrete concept has a life cycle"@en ;
  skos:prefLabel "Concrete concept"@en ;

.

sml:Entity
  a rdfs:Class ;
  rdfs:seeAlso sml-term:Entity ;
  rdfs:subClassOf sml:ConcreteConcept ;
  skos:definition "Concept that forms a manifestation and a demarcation
in a concrete space-time, and which has a certain state at any moment in
time. An entity has a unique identity that remains constant throughout
its life cycle. The life cycle of an entity is made up of the sequence of
states of that entity, which follow each other in time. An entity is an
object or an activity. An object exists, an activity takes place"@en ;
  skos:prefLabel "Entity"@en ;

.

sml:EnumerationType
  a rdfs:Class ;
  rdfs:seeAlso sml-term:EnumerationType ;
  rdfs:subClassOf rdfs:Container ;
  rdfs:subClassOf sml:Type ;
  skos:definition "A meta-concept having concepts as instances having no
further attributes or relations (annotations only)"@en ;
  skos:prefLabel "Enumeration type"@en ;

.

sml:Event
  a rdfs:Class ;
  rdfs:seeAlso sml-term:Event ;
  rdfs:subClassOf sml:ConcreteConcept ;
  skos:definition "Transition between two successive states of an entity
(object or activity). An event is triggered in a state"@en ;
  skos:prefLabel "Event"@en ;
  skos:prefLabel "Event"@en ;

.

sml:FunctionalEntity
  a rdfs:Class ;
  rdfs:seeAlso sml-term:FunctionalEntity ;
  rdfs:subClassOf sml:Entity ;

```

```

skos:definition "An entity involving the external behavior where the
output contributes to stakeholder objectives implemented/played by one or
more technical entities"@en ;
    skos:prefLabel "Functional entity"@en ;

.
sml:GeometricEntity
    a rdfs:Class ;
    rdfs:seeAlso sml-term:GeometricEntity ;
    rdfs:subClassOf sml:Representation ;
    skos:definition "Named concept, which forms an actual or virtual
demarcation in a concrete (physical, three-dimensional) space, which we
experience in reality"@en ;
    skos:prefLabel "Geometric entity"@en ;

.
sml:InformationObject
    a rdfs:Class ;
    rdfs:seeAlso sml-term:InformationObject ;
    rdfs:subClassOf sml:Object ;
    skos:definition "Object which describes a thing in reality"@en ;
    skos:prefLabel "Information object"@en ;

.
sml:Object
    a rdfs:Class ;
    rdfs:seeAlso sml-term:Object ;
    rdfs:subClassOf sml:Entity ;
    skos:definition "Entity that exists or can exist within a concrete
space-time. An object executes an activity, and is transformed by an
activity"@en ;
    skos:prefLabel "Object"@en ;

.
sml:Objectification
    a rdfs:Class ;
    rdfs:seeAlso sml-term:Objectification ;
    rdfs:subClassOf sml:AbstractConcept ;
    skos:definition "An attribute (quality or quantity) or relation
represented as an individual defined by a concept"@en ;
    skos:prefLabel "Objectification"@en ;

.
sml:PhysicalObject
    a rdfs:Class ;
    rdfs:seeAlso sml-term:PhysicalObject ;
    rdfs:subClassOf sml:Object ;
    skos:definition "Object that exists or may exist within physical 4D
space-time. A physical object forms a manifestation and a demarcation of
matter and/or energy, and is (in)directly perceptible by the senses"@en ;
    skos:prefLabel "Physical object"@en ;

.
sml:PlannedEntity
    a rdfs:Class ;
    rdfs:seeAlso sml-term:PlannedEntity ;
    rdfs:subClassOf sml:Entity ;
    skos:definition "An entity that does not yet exist in physical reality,
but which exists in mental or conceptual reality"@en ;
    skos:prefLabel "Planned entity"@en ;
.

```

**SS-EN 17632-1:2022 (E)**

```

sml:QualityValue
  a rdfs:Class ;
  rdfs:seeAlso sml-term:QualityValue ;
  rdfs:subClassOf sml:Objectification ;
  skos:definition "The objectification of a value of a quality having a complex value like a simple value sec combined with other metadata or just a combination of simple values"@en ;
  skos:prefLabel "Quality value"@en ;
.

sml:QuantityValue
  a rdfs:Class ;
  rdfs:seeAlso sml-term:QuantityValue ;
  rdfs:subClassOf sml:Objectification ;
  skos:definition "The objectification of a value of a quantity (typically involving a quantity kind and a unit) having a complex value like a simple value sec combined with other metadata or just a combination of simple values"@en ;
  skos:prefLabel "Quantity value"@en ;
.

sml:RealizedEntity
  a rdfs:Class ;
  rdfs:seeAlso sml-term:RealizedEntity ;
  rdfs:subClassOf sml:Entity ;
  skos:definition "An entity that exists or has existed in the physical reality"@en ;
  skos:prefLabel "Realized entity"@en ;
.

sml:RelationReference
  a rdfs:Class ;
  rdfs:seeAlso sml-term:RelationReference ;
  rdfs:subClassOf sml:Objectification ;
  skos:definition "The objectification of a reference for a relation having a complex value like a simple reference sec combined with other metadata or just a combination of simple references (n-ary relation)"@en ;
  skos:prefLabel "Relation reference"@en ;
.

sml:Representation
  a rdfs:Class ;
  rdfs:seeAlso sml-term:Representation ;
  rdfs:subClassOf sml:InformationObject ;
  skos:definition "That which represents something else"@en ;
  skos:prefLabel "Representation"@en ;
.

sml:State
  a rdfs:Class ;
  rdfs:seeAlso sml-term:State ;
  rdfs:subClassOf sml:ConcreteConcept ;
  skos:definition "Temporal part of an entity during a period between two events. A state is characterized by the properties and relations of the entity"@en ;
  skos:prefLabel "State"@en ;
.

sml:TechnicalEntity
  a rdfs:Class ;

```

```
rdfs:seeAlso sml-term:TechnicalEntity ;
rdfs:subClassOf sml:Entity ;
skos:definition "An entity concerned with technical properties and
relations that implements or plays functional entities"@en ;
skos:prefLabel "Technical entity"@en ;

.

sml:TopConcept
a rdfs:Class ;
rdfs:seeAlso sml-term:TopConcept ;
skos:definition "The most generic concept"@en ;
skos:prefLabel "Top concept"@en ;

.

sml:Type
a rdfs:Class ;
rdfs:seeAlso sml-term:Type ;
rdfs:subClassOf sml:AbstractConcept ;
skos:definition "A meta-concept which instances are concepts (not
individuals)"@en ;
skos:prefLabel "Type"@en ;

.

sml:abbreviation
a rdf:Property ;
rdfs:domain sml:TopConcept ;
rdfs:range xsd:string ;
rdfs:seeAlso sml-term:abbreviation ;
rdfs:subPropertyOf skos:altLabel ;
skos:definition "Acronym or initial word as special case of an external
name"@en ;
skos:prefLabel "abbreviation"@en ;

.

sml:begins
a rdf:Property ;
rdfs:domain sml:Event ;
rdfs:range sml:State ;
rdfs:seeAlso sml-term:begins ;
skos:definition "The state that is started by an event"@en ;
skos:prefLabel "begins"@en ;

.

sml:ends
a rdf:Property ;
rdfs:domain sml:Event ;
rdfs:range sml:State ;
rdfs:seeAlso sml-term:ends ;
skos:definition "The state that is ended by an event"@en ;
skos:prefLabel "ends"@en ;

.

sml:executes
a rdf:Property ;
rdfs:domain sml:PhysicalObject ;
rdfs:range sml:Activity ;
rdfs:seeAlso sml-term:executes ;
skos:definition "The activity executed by a physical object"@en ;
skos:prefLabel "executes"@en ;

.

sml:hasBoundary
```

**SS-EN 17632-1:2022 (E)**

```

a rdf:Property ;
rdfs:domain sml:ConcreteConcept ;
rdfs:range sml:GeometricEntity ;
rdfs:seeAlso sml-term:hasBoundary ;
skos:definition "The geometric entity representing the boundary of a
concrete concept"@en ;
skos:prefLabel "has boundary"@en ;
.

sml:hasFunctionalPart
a rdf:Property ;
rdfs:domain sml:FunctionalEntity ;
rdfs:range sml:FunctionalEntity ;
rdfs:seeAlso sml-term:hasFunctionalPart ;
rdfs:subPropertyOf sml:hasPart ;
skos:definition "The hasPart relation of a functional nature"@en ;
skos:prefLabel "has functional part"@en ;
.

sml:hasInterior
a rdf:Property ;
rdfs:domain sml:ConcreteConcept ;
rdfs:range sml:GeometricEntity ;
rdfs:seeAlso sml-term:hasInterior ;
skos:definition "The geometric entity representing the interior of a
concrete concept"@en ;
skos:prefLabel "has interior"@en ;
.

sml:hasPart
a rdf:Property ;
rdfs:domain sml:ConcreteConcept ;
rdfs:range sml:ConcreteConcept ;
rdfs:seeAlso sml-term:hasPart ;
skos:definition "A decomposition (hasPart) relation between concrete
concepts"@en ;
skos:prefLabel "has part"@en ;
.

sml:hasQuantityKind
a rdf:Property ;
rdfs:range qudt:QuantityKind ;
rdfs:seeAlso sml-term:hasQuantityKind ;
skos:definition "The possession of a quantity kind"@en ;
skos:prefLabel "has quantity kind"@en ;
.

sml:hasSolution
a rdf:Property ;
rdfs:domain sml:FunctionalEntity ;
rdfs:range sml:TechnicalEntity ;
rdfs:seeAlso sml-term:hasSolution ;
skos:definition "A technical entity that is a potential solution for a
functional entity"@en ;
skos:prefLabel "has solution"@en ;
.

sml:hasState
a rdf:Property ;
rdfs:domain sml:Entity ;
rdfs:range sml:State ;

```

```
rdfs:seeAlso sml-term:hasState ;
skos:definition "The state of an entity in a certain point or period of time"@en ;
skos:prefLabel "has state"@en ;

.

sml:hasTechnicalPart
a rdf:Property ;
rdfs:domain sml:TechnicalEntity ;
rdfs:range sml:TechnicalEntity ;
rdfs:seeAlso sml-term:hasTechnicalPart ;
rdfs:subPropertyOf sml:hasPart ;
skos:definition "A hasPart relation of a technical nature"@en ;
skos:prefLabel "has technical part"@en ;

.

sml:hasTemporalEntity
a rdf:Property ;
rdfs:domain sml:ConcreteConcept ;
rdfs:range time:TemporalEntity ;
rdfs:seeAlso sml-term:hasTemporalEntity ;
skos:definition "The abstract temporal entity representing a concrete concept in time"@en ;
skos:prefLabel "has temporal entity"@en ;

.

sml:hasUnit
a rdf:Property ;
rdfs:range qudt:Unit ;
rdfs:seeAlso sml-term:hasUnit ;
skos:definition "The possession of a unit"@en ;
skos:prefLabel "has unit"@en ;

.

sml:isDescribedBy
a rdf:Property ;
rdfs:domain sml:TopConcept ;
rdfs:range sml:InformationObject ;
rdfs:seeAlso sml-term:isDescribedBy ;
skos:definition "The information object that describes something"@en ;
skos:prefLabel "is described by"@en ;

.

sml:isImplementedBy
a rdf:Property ;
rdfs:domain sml:FunctionalEntity ;
rdfs:range sml:TechnicalEntity ;
rdfs:seeAlso sml-term:isImplementedBy ;
rdfs:subPropertyOf sml:hasSolution ;
skos:definition "The technical entity that implements a functional entity"@en ;
skos:prefLabel "is implemented by"@en ;

.

sml:isPlanFor
a rdf:Property ;
rdfs:domain sml:PlannedEntity ;
rdfs:range sml:RealizedEntity ;
rdfs:seeAlso sml-term:isPlanFor ;
skos:definition "A potentially realized entity according to this planned entity"@en ;
```

**SS-EN 17632-1:2022 (E)**

```

skos:prefLabel "is plan for"@en ;
.

sml:isRealizedBy
a rdf:Property ;
rdfs:domain sml:PlannedEntity ;
rdfs:range sml:RealizedEntity ;
rdfs:seeAlso sml-term:isRealizedBy ;
skos:definition "The realized entity that realizes a planned entity"@en
;
rdfs:subPropertyOf sml:isPlanFor ;
skos:prefLabel "is realized by"@en ;
.

sml:transforms
a rdf:Property ;
rdfs:domain sml:Activity ;
rdfs:range sml:Object ;
rdfs:seeAlso sml-term:transforms ;
skos:definition "The object transformed by an activity"@en ;
skos:prefLabel "transforms"@en ;
.

sml:triggers
a rdf:Property ;
rdfs:domain sml:Entity ;
rdfs:range sml:Event ;
rdfs:seeAlso sml-term:triggers ;
skos:definition "The event triggered by an entity"@en ;
skos:prefLabel "triggers"@en ;
.
```

**A.4 OWL part**

```

@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 sh: <http://www.w3.org/ns/shacl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sml: <https://w3id.org/sml/def#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/sml/owl/def>
a 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> ;
owl:imports <http://www.w3.org/2006/time> ;
owl:imports <http://www.w3.org/ns/shacl> ;
owl:imports <https://w3id.org/sml/rdfs/def> ;
.

sml:AbstractConcept
.
```

```
a owl:Class ;  
.sml:Activity  
a owl:Class ;  
rdfs:subClassOf [  
    a owl:Restriction ;  
    owl:allValuesFrom sml:Activity ;  
    owl:onProperty sml:hasPart ;  
] ;  
.sml:ConceptType  
a owl:Class ;  
.sml:ConcreteConcept  
a owl:Class ;  
.sml:Entity  
a owl:Class ;  
.sml:EnumerationType  
a owl:Class ;  
.sml:Event  
a owl:Class ;  
rdfs:subClassOf [  
    a owl:Restriction ;  
    owl:allValuesFrom sml:Event ;  
    owl:onProperty sml:hasPart ;  
] ;  
.sml:FunctionalEntity  
a owl:Class ;  
rdfs:subClassOf [  
    a owl:Restriction ;  
    owl:allValuesFrom sml:Entity ;  
    owl:onProperty sml:hasPart ;  
] ;  
.sml:GeometricEntity  
a owl:Class ;  
.sml:InformationObject  
a owl:Class ;  
.sml:Object  
a owl:Class ;  
rdfs:subClassOf [  
    a owl:Restriction ;  
    owl:allValuesFrom sml:Object ;  
    owl:onProperty sml:hasPart ;  
] ;  
.sml:Objectification  
a owl:Class ;  
rdfs:subClassOf [
```

**SS-EN 17632-1:2022 (E)**

```

    a owl:Restriction ;
    owl:maxCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty rdf:value ;
] ;

sml:PhysicalObject
a owl:Class ;

sml:PlannedEntity
a owl:Class ;
rdfs:subClassOf [
    a owl:Restriction ;
    owl:allValuesFrom sml:PlannedEntity ;
    owl:onProperty sml:hasPart ;
] ;

sml:QualityValue
a owl:Class ;
rdfs:subClassOf [
    a owl:Restriction ;
    owl:allValuesFrom xsd:string ;
    owl:onProperty rdf:value ;
] ;

sml:QuantityValue
a owl:Class ;
rdfs:subClassOf [
    a owl:Restriction ;
    owl:allValuesFrom [
        a owl:Class ;
        owl:unionOf (
            xsd:decimal
            xsd:float
            xsd:double
        ) ;
    ] ;
    owl:onProperty rdf:value ;
] ;

sml:RealizedEntity
a owl:Class ;
rdfs:subClassOf [
    a owl:Restriction ;
    owl:allValuesFrom sml:RealizedEntity ;
    owl:onProperty sml:hasPart ;
] ;

sml:RelationReference
a owl:Class ;
rdfs:subClassOf [
    a owl:Restriction ;
    owl:allValuesFrom xsd:anyURI ;
    owl:onProperty rdf:value ;
] ;

```

```
sml:Representation
  a owl:Class ;
.

sml:State
  a owl:Class ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:allValuesFrom sml:State ;
    owl:onProperty sml:hasPart ;
  ] ;

.

sml:TechnicalEntity
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:allValuesFrom sml:Entity ;
    owl:onProperty sml:hasPart ;
  ] ;

.

sml:TopConcept
  a owl:Class ;
.

sml:Type
  a owl:Class ;
.

sml:abbreviation
  a owl:AnnotationProperty ;
.

sml:begins
  a owl:ObjectProperty ;
.

sml:ends
  a owl:ObjectProperty ;
.

sml:hasBoundary
  a owl:ObjectProperty ;
.

sml:hasFunctionalPart
  a owl:ObjectProperty ;
.

sml:hasInterior
  a owl:ObjectProperty ;
.

sml:hasPart
  a owl:ObjectProperty ;
.

sml:hasQuantityKind
  a owl:ObjectProperty ;
.

sml:hasSolution
  a owl:ObjectProperty ;
.

sml:hasState
  a owl:ObjectProperty ;
.

sml:hasTechnicalPart
```

**SS-EN 17632-1:2022 (E)**

```
a owl:ObjectProperty ;  
.  
sml:hasTemporalEntity  
    a owl:ObjectProperty ;  
.  
sml:hasUnit  
    a owl:ObjectProperty ;  
.  
sml:isDescribedBy  
    a owl:ObjectProperty ;  
.  
sml:isImplementedBy  
    a owl:ObjectProperty ;  
.  
sml:isPlanFor  
    a owl:ObjectProperty ;  
.  
sml:isRealizedBy  
    a owl:ObjectProperty ;  
.  
sml:executes  
    a owl:ObjectProperty ;  
.  
sml:transforms  
    a owl:ObjectProperty ;  
.  
sml:triggers  
    a owl:ObjectProperty ;  
.  
sml:AllDisjointClasses_1  
    a owl:AllDisjointClasses ;  
    owl:members (  
        sml:PhysicalObject  
        sml:InformationObject  
        sml:Activity  
        sml:Event  
        sml:State  
    ) ;  
.  
sml:AllDisjointClasses_2  
    a owl:AllDisjointClasses ;  
    owl:members (  
        sml:FunctionalEntity  
        sml:TechnicalEntity  
    ) ;  
.  
sml:AllDisjointClasses_3  
    a owl:AllDisjointClasses ;  
    owl:members (  
        sml:PlannedEntity  
        sml:RealizedEntity  
    ) ;  
.
```

## A.5 SHACL part

```

@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 sh: <http://www.w3.org/ns/shacl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sml: <https://w3id.org/sml/def#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/sml/shacl>
  a 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> ;
  owl:imports <http://www.w3.org/2006/time> ;
  owl:imports <http://www.w3.org/ns/shacl> ;
  owl:imports <https://w3id.org/sml/rdfs/def> ;
.

sml:Activity
  a sh:NodeShape ;
  sh:property [
    sh:path sml:hasBoundary ;
    sh:class sml:GeometricEntity ;
  ] ;
  sh:property [
    sh:path sml:hasInterior ;
    sh:class sml:GeometricEntity ;
  ] ;
  sh:property [
    sh:path sml:hasPart ;
    sh:class sml:Activity ;
  ] ;
  sh:property [
    sh:path sml:hasPeriod ;
    sh:class time:TemporalEntity ;
  ] ;
  sh:property [
    sh:path sml:transforms ;
    sh:class sml:Object ;
  ] ;
.

sml:ConcreteConcept
  a sh:NodeShape ;
  sh:property [
    sh:path sml:hasBoundary ;
    sh:class sml:GeometricEntity ;
  ] ;
  sh:property [

```

**SS-EN 17632-1:2022 (E)**

```
sh:path sml:hasInterior ;
sh:class sml:GeometricEntity ;
] ;
sh:property [
    sh:path sml:hasTemporalEntity ;
    sh:class time:TemporalEntity ;
] ;
sh:property [
    sh:path sml:hasPart ;
    sh:class sml:ConcreteConcept ;
] ;
.

sml:Entity
a sh:NodeShape ;
sh:property [
    sh:path sml:hasState ;
    sh:class sml:State ;
] ;
sh:property [
    sh:path sml:triggers ;
    sh:class sml:Event ;
] ;
.

sml:Event
a sh:NodeShape ;
sh:property [
    sh:path sml:begins ;
    sh:class sml:State ;
] ;
sh:property [
    sh:path sml:ends ;
    sh:class sml:State ;
] ;
sh:property [
    sh:path sml:hasPart ;
    sh:class sml:Event ;
] ;
.

sml:FunctionalEntity
a sh:NodeShape ;
sh:property [
    sh:path sml:hasFunctionalPart ;
    sh:class sml:FunctionalEntity ;
] ;
sh:property [
    sh:path sml:hasSolution ;
    sh:class sml:TechnicalEntity ;
] ;
sh:property [
    sh:path sml:hasPart ;
    sh:class sml:Entity ;
] ;
.

sml:Object
a sh:NodeShape ;
```

```
sh:property [
    sh:path sml:hasPart ;
    sh:class sml:Object ;
] ;

sml:Objectification
a sh:NodeShape ;
sh:property [
    sh:path rdf:value;
    sh:maxCount 1;
] ;

sml:PhysicalObject
a sh:NodeShape ;
sh:property [
    sh:path sml:isDescribedBy ;
    sh:class sml:InformationObject ;
] ;
sh:property [
    sh:path sml:executes ;
    sh:class sml:Activity ;
] ;

sml:PlannedEntity
a sh:NodeShape ;
sh:property [
    sh:path sml:hasPart ;
    sh:class sml:PlannedEntity ;
] ;
sh:property [
    sh:path sml:isPlanFor ;
    sh:class sml:RealizedEntity ;
] ;

sml:QualityValue
a sh:NodeShape ;
sh:property [
    sh:path rdf:value ;
    sh:datatype xsd:string ;
] ;

sml:QuantityValue
a sh:NodeShape ;
sh:property [
    sh:path rdf:value ;
    sh:or ( [ sh:datatype xsd:decimal ] [sh:datatype xsd:float] [
sh:datatype xsd:double ] );
] ;

sml:RealizedEntity
a sh:NodeShape ;
sh:property [
    sh:path sml:hasPart ;
    sh:class sml:RealizedEntity ;
]
```

**SS-EN 17632-1:2022 (E)**

```

.
sml:RelationReference
  a sh:NodeShape ;
  sh:property [
    sh:path rdf:value ;
    sh:datatype xsd:anyURI ;
  ] ;

.
sml:State
  a sh:NodeShape ;
  sh:property [
    sh:path sml:hasBoundary ;
    sh:class sml:GeometricEntity ;
  ] ;
  sh:property [
    sh:path sml:hasInterior ;
    sh:class sml:GeometricEntity ;
  ] ;
  sh:property [
    sh:path sml:hasPart ;
    sh:class sml:State ;
  ] ;
  sh:property [
    sh:path sml:hasPeriod ;
    sh:class time:TemporalEntity ;
  ] ;

.
sml:TechnicalEntity
  a sh:NodeShape ;
  sh:property [ sh:path sml:hasTechnicalPart ;
  sh:class sml:TechnicalEntity ; ] ;
  sh:property [
    sh:path sml:hasPart ;
    sh:class sml:Entity ;
  ] ;

.
sml:TopConcept
  a sh:NodeShape ;
  sh:property [
    sh:path sml:abbreviation ;
    sh:datatype xsd:string ;
  ] ;
  sh:property [
    sh:path sml:isDescribedBy ;
    sh:class sml:InformationObject ;
  ] ;

.
sml:hasQuantityKindShape
  a sh:NodeShape ;
  sh:class qudt:QuantityKind ;
  sh:targetObjectsOf sml:hasQuantityKind ;

.
sml:hasUnitShape
  a sh:NodeShape ;
  sh:class qudt:Unit ;

```

```
sh:targetObjectsOf sml:hasUnit ;  
. . .  
sml:AllDisjointClassesShape  
a sh:NodeShape ;  
sh:targetSubjectsOf rdf:type ;  
sh:property [  
    sh:path (rdf:type [sh:zeroOrMorePath rdfs:subClassOf]) ;  
    sh:qualifiedValueShape [  
        sh:in (  
            sml:PhysicalObject  
            sml:InformationObject  
            sml:State  
            sml:Event  
            sml:Activity  
        ) ;  
    ] ;  
    sh:qualifiedMaxCount 1 ;  
] ;  
sh:property [  
    sh:path (rdf:type [sh:zeroOrMorePath rdfs:subClassOf]) ;  
    sh:qualifiedValueShape [  
        sh:in (  
            sml:FunctionalEntity  
            sml:TechnicalEntity  
        ) ;  
    ] ;  
    sh:qualifiedMaxCount 1 ;  
] ;  
sh:property [  
    sh:path (rdf:type [sh:zeroOrMorePath rdfs:subClassOf]) ;  
    sh:qualifiedValueShape [  
        sh:in (  
            sml:PlannedEntity  
            sml:RealizedEntity  
        ) ;  
    ] ;  
    sh:qualifiedMaxCount 1 ;  
] ;  
. . .
```

**SS-EN 17632-1:2022 (E)**

**Annex B**  
(normative)

**Selected W3C RDF language subsets**

**B.1 General**

This annex describes the exact language constructs subsets used by this document for the various Levels of Capability (XSD, RDF, SKOS, RDFS, OWL and SHACL).

**B.2 XML schema (XSD), part 2: Datatypes 2<sup>nd</sup> edition**

- xsd:string
- xsd:integer
- xsd:boolean
- xsd:decimal
- xsd:float
- xsd:double
- xsd:anyURI
- xsd:date
- xsd:time
- xsd:dateTime
- xsd:duration
- xsd:minInclusive
- xsd:maxInclusive
- xsd:minExclusive
- xsd:maxExclusive

**B.3 Resource description framework (RDF)**

- rdf:type
- rdf:value
- rdf:Statement
- rdf:subject

- rdf:predicate
- rdf:object

#### B.4 Simple knowledge organization system (SKOS)

- skos:ConceptScheme
- skos:inScheme
- skos:Concept
- skos:Collection
- skos:member
- skos:definition
- skos:notation
- skos:example
- skos:scopeNote
- skos:prefLabel
- skos:altLabel
- skos:editorialNote
- skos:topConceptOf
- skos:narrower
- skos:broader
- skos:exactMatch
- skos:closeMatch
- skos:narrowMatch
- skos:broadMatch
- skos:relatedMatch

#### B.5 Resource description framework schema (RDFS)

- rdfs:Class
- rdfs:Datatype
- rdfs:subClassOf

## SS-EN 17632-1:2022 (E)

- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range
- rdfs:label
- rdfs:comment
- rdfs:isDefinedBy
- rdfs:seeAlso
- rdfs:Container
- rdfs:member

## B.6 Web ontology language (OWL)

- owl:Ontology
- owl:imports
- owl:Class
- owl:DatatypeProperty
- owl:ObjectProperty
- owl:AnnotationProperty
- owl:inverseOf
- owl:FunctionalProperty
- owl:InverseFunctionalProperty
- owl:oneOf
- owl:unionOf
- owl:intersectionOf
- owl:disjointWith
- owl:AllDisjointClasses
- owl:Restriction
- owl:allValuesFrom
- owl:someValuesFrom
- owl:hasValue

- owl:minCardinality
- owl:maxCardinality
- owl:cardinality
- owl:minQualifiedCardinality
- owl:maxQualifiedCardinality
- owl:qualifiedCardinality
- owl:onProperty
- owl:onDatatype
- owl:onClass
- owl:withRestrictions
- owl:propertyChainAxiom
- owl:Thing
- owl:topDataProperty
- owl:topObjectProperty
- owl:equivalentClass
- owl:equivalentProperty
- owl:sameAs

## B.7 Shape constraint language (SHACL)

- sh:NodeShape
- sh:PropertyShape
- sh:property
- sh:path
- sh:inversePath
- sh:targetNode
- sh:targetClass
- sh:targetSubjectsOf
- sh:targetObjectsOf
- sh:nodeKind

## SS-EN 17632-1:2022 (E)

- sh:BlankNode
- sh:IRI
- sh:Literal
- sh:BlankNodeOrIRI
- sh:BlankNodeOrLiteral
- sh:IRIOrLiteral
- sh:datatype
- sh:class
- sh:minCount
- sh:maxCount
- sh:qualifiedMinCount
- sh:qualifiedMaxCount
- sh:qualifiedValueShape
- sh:qualifiedValueShapesDisjoint
- sh:not
- sh:or
- sh:xone
- sh:and
- sh:languageIn
- sh:uniqueLang
- sh:pattern
- sh:name
- sh:description
- sh:group
- sh:PropertyGroup
- sh:order
- sh:defaultValue
- sh:severity

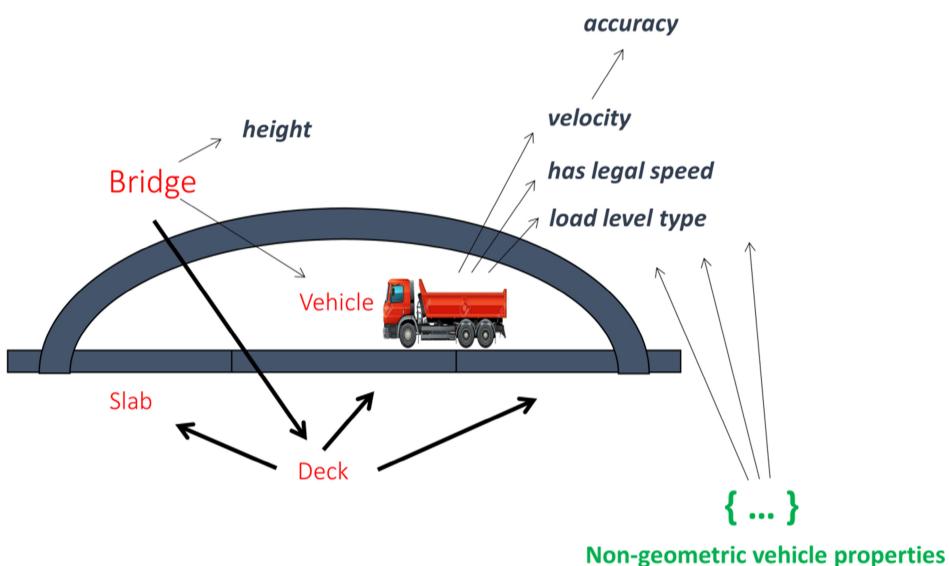
- sh:Warning
- sh:Info
- sh:Violation
- sh:in
- sh:closed
- sh:ignoredProperties
- sh:minInclusive
- sh:maxInclusive
- sh:minExclusive
- sh:maxExclusive
- sh:declare
- sh:prefix
- sh:namespace
- sh:sparql
- sh:SPARQLConstraint
- sh:select
- sh:message
- sh:prefixes
- sh:entailment
- sh:zeroOrMorePath

## Annex C (informative)

### SML Example in SKOS/RDFS/OWL/SHACL (Turtle format)

#### C.1 Example description

Bridges are special kind of physical objects. Bridges always have exactly one height (a length quantity kind) expressed in meters. Bridges always consist of exactly one bridge deck that itself consists of one or more concrete slabs. At a certain moment in time there are zero or more vehicles on a bridge with a certain velocity (a speed quantity kind) measured in kilometers per hour (with an accuracy of 90%) having a certain load level type, which can only be selected from light, medium or heavy. It is known for each vehicle whether they are complying with the applicable legal speed limit at that time. Velocity, has legal speed and load level type are grouped as non-geometric vehicle properties.



**Figure C.1 — Graphical example description**

#### C.2 SKOS part

```

@prefix ex-term: <https://w3id.org/sml-example/term#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix qudt: <http://qudt.org/schema/qudt/> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix sml-term: <https://w3id.org/sml/term#> .

```

```
<https://w3id.org/sml-example/skos/term>
  a skos:ConceptScheme ;
  owl:imports <https://w3id.org/sml/skos/term> ;

  ex-term:Bridge
    a skos:Concept ;
    skos:prefLabel "Bridge"@en ;
    skos:definition "A structure carrying a road, path, railway, etc.
across a river, road, or other obstacle"@en ;
    skos:broader sml-term:PhysicalObject ;

  ex-term:Deck
    a skos:Concept ;
    skos:prefLabel "Deck"@en ;
    skos:definition "The surface of a bridge"@en ;
    skos:broader sml-term:PhysicalObject ;

  ex-term:Heavy
    a skos:Concept ;
    skos:prefLabel "Heavy"@en ;
    skos:broader ex-term:LoadLevelType ;

  ex-term:Light
    a skos:Concept ;
    skos:prefLabel "Light"@en ;
    skos:broader ex-term:LoadLevelType ;

  ex-term:LoadLevelType
    a skos:Concept ;
    skos:prefLabel "Load level type"@en ;
    skos:broader sml-term:EnumerationType ;

  ex-term:NonGeometricProperties
    a skos:Concept ;
    skos:prefLabel "Non-geometric properties"@en ;

  ex-term:Normal
    a skos:Concept ;
    skos:prefLabel "Normal"@en ;
    skos:broader ex-term:LoadLevelType ;

  ex-term:Slab
    a skos:Concept ;
    skos:prefLabel "Slab"@en ;
    skos:definition "A large, thick, flat piece of stone or concrete,
typically square or rectangular in shape"@en ;
    skos:broader sml-term:PhysicalObject ;

  ex-term:Vehicle
    a skos:Concept ;
    skos:prefLabel "Vehicle"@en ;
    skos:definition "A thing used for transporting people or goods,
especially on land, such as a car, lorry, or cart"@en ;

  ex-term:accuracy
```

**SS-EN 17632-1:2022 (E)**

```

a skos:Concept ;
  skos:prefLabel "accuracy"@en ;

.
ex-term:currentlyServingVehicle
a skos:Concept ;
  skos:prefLabel "currently serving vehicle"@en ;

.
ex-term:hasLegalSpeed
a skos:Concept ;
  skos:prefLabel "has legal speed"@en ;

.
ex-term:height
a skos:Concept ;
  skos:prefLabel "height"@en ;

.
ex-term:loadLevelType
a skos:Concept ;
  skos:prefLabel "load level type"@en ;

.
ex-term:velocity
a skos:Concept ;
  skos:prefLabel "velocity"@en ;
.
```

**C.3 RDFS part**

**NOTE** The typing to RDFS classes is not really needed since it can be inferred from the typing to OWL classes. For clarity and potential separate use they are made explicit.

```

@prefix ex: <https://w3id.org/sml-example/def#> .
@prefix ex-term: <https://w3id.org/sml-example/term#> .
@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 sh: <http://www.w3.org/ns/shacl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sml: <https://w3id.org/sml/def#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/sml-example/rdfs/def>
  a owl:Ontology ;
  owl:imports <https://w3id.org/sml/rdfs/def> ;

.
ex:Bridge
a rdfs:Class ;
  rdfs:seeAlso ex-term:Bridge ;
  rdfs:subClassOf sml:PhysicalObject ;
  skos:definition "A structure carrying a road, path, railway, etc. across a river, road, or other obstacle"@en ;
  skos:prefLabel "Bridge"@en ;
.
```

```
ex:Deck
  a rdfs:Class ;
  rdfs:seeAlso ex-term:Deck ;
  rdfs:subClassOf sml:PhysicalObject ;
  skos:definition "The surface of a bridge"@en ;
  skos:prefLabel "Deck"@en ;

.
ex:Heavy
  a ex:LoadLevelType ;
  rdfs:seeAlso ex-term:Heavy ;
  skos:prefLabel "Heavy"@en ;

.
ex:Light
  a ex:LoadLevelType ;
  rdfs:seeAlso ex-term:Light ;
  skos:prefLabel "Light"@en ;

.
ex:LoadLevelType
  a rdfs:Class ;
  rdfs:seeAlso ex-term:LoadLevelType ;
  a sml:EnumerationType ;
  skos:prefLabel "Load level type"@en ;

.
ex:NonGeometricProperties
  a rdfs:Container ;
  rdfs:seeAlso ex-term:NonGeometricProperties ;
  rdfs:member ex:loadLevelType ;
  rdfs:member ex:velocity ;
  rdfs:member ex:hasLegalSpeed ;
  skos:prefLabel "Non-geometric properties"@en ;

.
ex:Normal
  a ex:LoadLevelType ;
  rdfs:seeAlso ex-term:Normal ;
  skos:prefLabel "Normal"@en ;

.
ex:Slab
  a rdfs:Class ;
  rdfs:seeAlso ex-term:Slab ;
  rdfs:subClassOf sml:PhysicalObject ;
  skos:definition "A large, thick, flat piece of stone or concrete,
typically square or rectangular in shape"@en ;
  skos:prefLabel "Slab"@en ;

.
ex:Vehicle
  a rdfs:Class ;
  rdfs:seeAlso ex-term:Vehicle ;
  rdfs:subClassOf sml:PhysicalObject ;
  skos:definition "A thing used for transporting people or goods,
especially on land, such as a car, lorry, or cart"@en ;
  skos:prefLabel "Vehicle"@en ;

.
ex:accuracy
  a rdf:Property ;
  rdfs:seeAlso ex-term:accuracy ;
```

**SS-EN 17632-1:2022 (E)**

```
rdfs:range sml:QuantityValue ;
skos:prefLabel "accuracy"@en ;

.

ex:currentlyServingVehicle
a rdf:Property ;
rdfs:seeAlso ex-term:currentlyServingVehicle ;
rdfs:range ex:Vehicle ;
skos:prefLabel "currently serving vehicle"@en ;

.

ex:hasLegalSpeed
a rdf:Property ;
rdfs:domain ex:Vehicle ;
rdfs:seeAlso ex-term:hasLegalSpeed ;
rdfs:range xsd:boolean ;
skos:prefLabel "has legal speed"@en ;

.

ex:height
a rdf:Property ;
rdfs:seeAlso ex-term:height ;
rdfs:range sml:QuantityValue ;
sml:quantityKind quantitykind:Length ;
skos:prefLabel "height"@en ;

.

ex:loadLevelType
a rdf:Property ;
rdfs:domain ex:Vehicle ;
rdfs:seeAlso ex-term:loadLevelType ;
rdfs:range ex:LoadLevelType ;
skos:prefLabel "load level type"@en ;

.

ex:velocity
a rdf:Property ;
rdfs:seeAlso ex-term:velocity ;
rdfs:range sml:QuantityValue ;
sml:quantityKind quantitykind:Speed ;
skos:prefLabel "velocity"@en ;
```

## C.4 OWL part

```

@prefix ex: <https://w3id.org/sml-example/def#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix qudt: <http://qudt.org/schema/qudt/> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix sml: <http://w3id.org/sml/def#> .

<https://w3id.org/sml-example/owl/def>
  a owl:Ontology ;
  owl:imports <https://w3id.org/sml/owl/def> ;
  owl:imports <https://w3id.org/sml-example/rdfs/def> ;
.

ex:Bridge
  a owl:Class ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty ex:height ;
  ] ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onClass ex:Deck ;
    owl:onProperty sml:hasPart ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
.

ex:Deck
  a owl:Class ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onClass ex:Slab ;
    owl:onProperty sml:hasPart ;
  ] ;
.

ex:LoadLevelType
  a owl:Class ;
  owl:oneOf (
    ex:Light
    ex:Normal
    ex:Heavy
  ) ;
.

ex:Slab
  a owl:Class ;
.

ex:Vehicle
.
```

**SS-EN 17632-1:2022 (E)**

```

a owl:Class ;
.

ex:accuracy
a owl:ObjectProperty ;
.

ex:currentlyServingVehicle
a owl:ObjectProperty ;
.

ex:hasLegalSpeed
a owl:DatatypeProperty ;
.

ex:height
a owl:ObjectProperty ;
.

ex:loadLevel
a owl:ObjectProperty ;
.

ex:velocity
a owl:ObjectProperty ;
.
```

**C.5 SHACL part**

```

@prefix ex: <https://w3id.org/sml-example/def#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix qudt: <http://qudt.org/schema/qudt> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix quantitykind: <http://qudt.org/vocab/quantitykind> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix sml: <http://w3id.org/sml/def#> .

<https://w3id.org/sml-example/shacl/def>
a owl:Ontology ;
owl:imports <https://w3id.org/sml/shacl/def> ;
owl:imports <https://w3id.org/sml-example/rdfs/def> ;
owl:imports <http://www.w3.org/ns/shacl> ;
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/2006/time> ;

ex:Bridge
a sh:NodeShape ;
sh:property [
  sh:path ex:height ;
  sh:minCount 1 ;
]
sh:property [
  sh:path sml:hasPart ;
  sh:minCount 1 ;
]
```

```

sh:maxCount 1 ;
sh:class ex:Deck ;
] ;

ex:Deck
a sh:NodeShape ;
sh:property [
sh:path sml:hasPart ;
sh:minCount 1 ;
sh:class ex:Slab ;
] ;

ex:Vehicle
a sh:NodeShape ;
sh:property [
sh:path ex:loadLevel ;
sh:class ex:LoadLevelType ;
sh:in (ex:Light ex:Normal ex:Heavy ) ;
] ;
sh:property [
sh:path ex:hasLegalSpeed ;
sh:datatype xsd:boolean ;
] ;

ex:accuracy
a sh:NodeShape ;
sh:class sml:QuantityValue ;

ex:currentlyServingVehicle
a sh:NodeShape ;
sh:class ex:Vehicle ;

ex:height
a sh:NodeShape ;
sh:class sml:QuantityValue ;

ex:velocity
a sh:NodeShape ;
sh:class sml:QuantityValue ;

```

## C.6 Data part

```

@prefix exdata: <https://w3id.org/sml-example/id#> .
@prefix ex: <https://w3id.org/sml-example/def#> .
@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 sml: <https://w3id.org/sml/def#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

```

**SS-EN 17632-1:2022 (E)**

```
<https://w3id.org/sml-example/id>
  a owl:Ontology ;
  owl:imports <https://w3id.org/sml-example/skos/term> ;
  owl:imports <https://w3id.org/sml-example/rdfs/def> ;
  owl:imports <https://w3id.org/sml-example/owl/def> ;
  owl:imports <https://w3id.org/sml-example/shacl/def> ;
  .

exdata:Bridge_1
  a ex:Bridge ;
  sml:hasPart exdata:Deck_1 ;
  ex:currentlyServingVehicle exdata:Vehicle_1 ;
  ex:height [
    rdf:value "50.0"^^xsd:double ;
    sml:unit unit:M ;
  ] ;
  .

exdata:Deck_1
  a ex:Deck ;
  sml:hasPart exdata:Slab_1 ;
  sml:hasPart exdata:Slab_2 ;
  sml:hasPart exdata:Slab_3 ;
  .

exdata:Slab_1
  a ex:Slab ;
  .

exdata:Slab_2
  a ex:Slab ;
  .

exdata:Slab_3
  a ex:Slab ;
  .

exdata:Vehicle_1
  a ex:Vehicle ;
  ex:hasLegalSpeed true ;
  ex:loadLevel ex:Heavy ;
  ex:velocity [
    rdf:value "128.0"^^xsd:double ;
    sml:unit unit:KiloM-PER-HR ;
    ex:accuracy [
      rdf:value "0.9"^^xsd:double ;
    ] ;
  ] ;
  .
```

## Annex D (informative)

### **Relationships with other asset/product modelling standards**

#### **D.1 General**

This annex describes the relationships between the SML standard and three directly related standards: the EN ISO 21597 series (Information container for linked document delivery), EN ISO 23387 (Building information modelling (BIM) - Data templates for construction objects used in the life cycle of built assets - Concepts and principles) and the ISO 15926 series (Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities).

#### **D.2 Relationship with the ISO 21597 series**

EN ISO 21597-1 [13], Information container for linked document delivery (ICDD) [13], defines a standard 'linked data'-based way to package interrelated documents for mostly human interpretation.

It specifies an information container with a payload of documents and objectified links (links modelled as a Class) between those documents or elements within those documents (the so-called "deep links").

In principle, ICDD only focuses on information exchange (referred to as "delivery"), not on information sharing. However, both the information container description and the document and link descriptions could also be reused in the context of information sharing (like for publication of a package of information on a web site in the cloud).

ICDD also does not focus specifically on LD/SW in the payload (just uses LD/SW technology for description OF the payload) but more on a hybrid situation of unstructured and structured documents (including possibly 'linked data'-based information sets and information models).

EN ISO 21597-2 [14] offers more meaningful/semantic specializations for the generic document links from part 1.

SML and ICDD can to a large extent complement each other. Known overlaps are:

- ICDD two-way deep linking would actually have same effect as linking in SML; and
- The ICDD Part 1 "ct:Document" is similar to the SML Top Level "sml:InformationObject".

NOTE      Although ICDD assumes CWA, it uses primarily default-OWA-based OWL.

#### **D.3 Relationship with EN ISO 23387**

##### **D.3.1 Introduction**

According to EN ISO 23387 [16], data templates are instances of the EN ISO 12006-3 (meta)schema in EXPRESS or XML schema definition (XSD), formatted in STEP physical file format (SPFF) respectively extensible markup language (XML). Actual instantiations of data templates are dealt with via references from IFC (SPFF or XML) files according to EN ISO 16739-1 [11].

According to this document, data templates are ontologies in (RDF/)RDFS/OWL/SHACL (depending on the use case type), formatted in a linked data format like Turtle, that can be instantiated again in Turtle information set files.

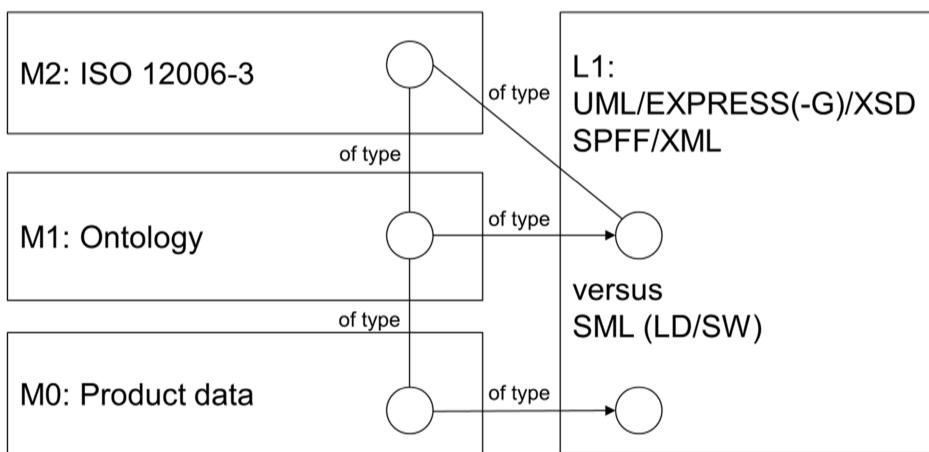
## SS-EN 17632-1:2022 (E)

Key advantages using the approach in this document over the EN ISO 12006-3 approach include:

- Modernized underlying technology: from ISO STEP/W3C XML-technology to fully web-based W3C ‘linked data’ technology;
- Standard direct (potentially distributed) data access via the SPARQL query language instead of platform-specific Application Programming Interfaces (APIs) avoiding software vendor/content provider lock-in;
- Increased modelling power for advanced information modelling involving i.e. complex constraints and derivations;
- Standard mechanisms for (inter)linking (elements) on information set level and information model level and linking between (elements of) information sets and information models; and
- bSI IFC-independent asset/product information instantiation.

### D.3.2 Modelling relationship with EN ISO 12006-3

Using the Figure 1 from Clause 1 (Scope), Figure D.1 shows that EN ISO 12006-3 (on M2) is conceptually complementary to the scope of this document (being M1/M0).



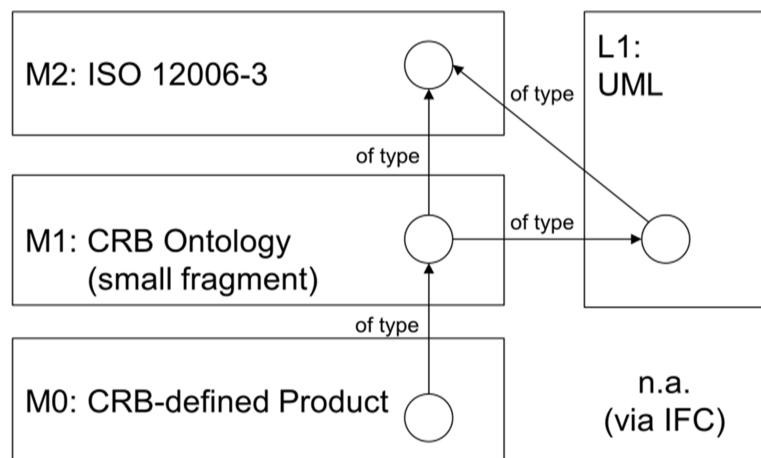
**Figure D.1 — EN ISO 12006-3 versus SML**

The EN ISO 12006-3 schema is a M2: Meta-‘Information Model’ on top of a M1: Ontology. Formulated the other way round, the instances of EN ISO 12006-3 meta-concepts are ontology elements.

The main difference is in the underlying L1 information language (EN ISO 12006-3 using OMG UML /ISO STEP/W3C XML technology, this document using W3C LD/SW technology). This means the items (circles) on the left are typed towards different language constructs (circles) on the right.

### D.3.3 Example Schweizerische Zentralstelle für Baurationalisation (CRB)

This example starts with an UML variant of a very small CRB ontology fragment as instantiation of EN ISO 12006-3 (all in UML) in Figure D.2 and Figure D.3.



**Figure D.2 — CRB example according to EN ISO 12006-3**

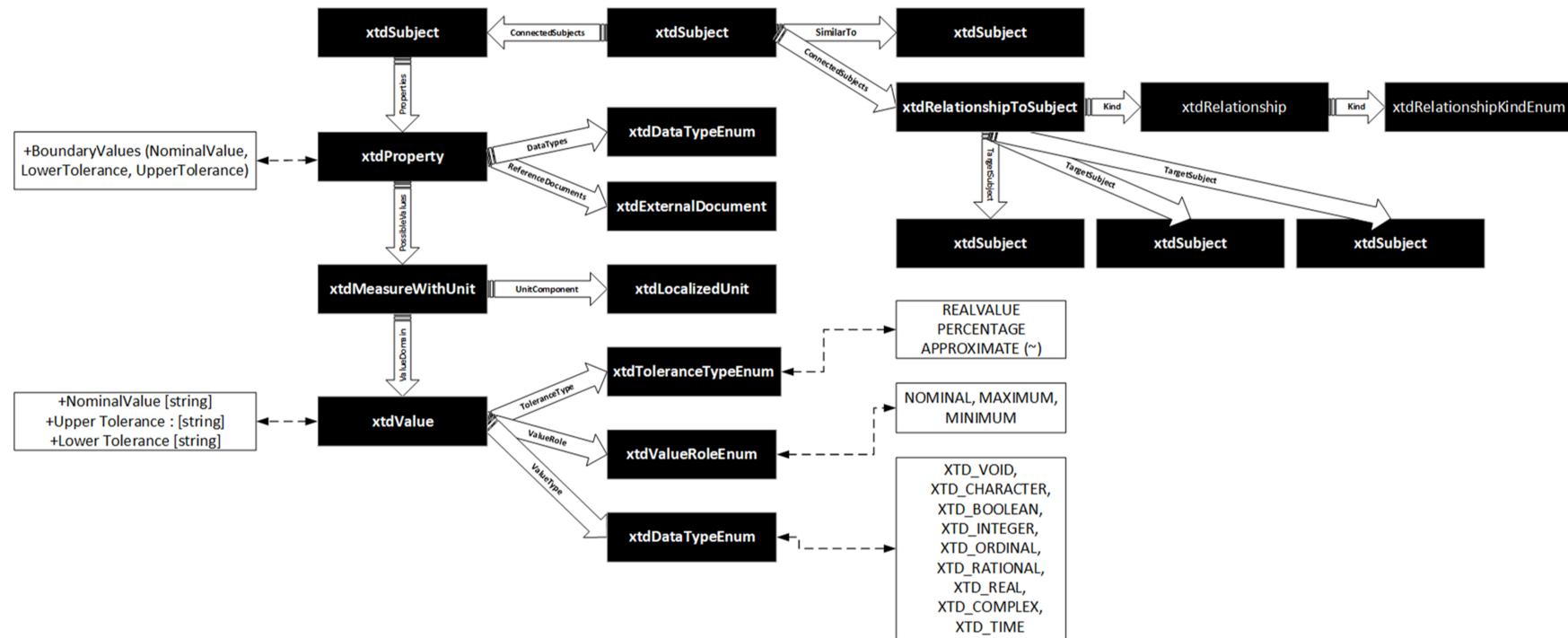


Figure D.3 —Relevant EN ISO 12006-3 fragment in UML-like format

In the example there is a CRB “door” that is specialized into an “interior door” and an “exterior door” all having a relevant “maximum pressure” represented as a “Crb\_Measure\_Pressure\_kn\_m2” quantity (Figure D.4). This quantity is measured in “kN/m<sup>2</sup>” and is defined by a reference document called “NPK622V19\_D\_200\_00012”.

An “external door” is further classified due to a standard constraint or minimum requirement of acceptable wind resistance in conjunction to CRB cost standards. This subclass of external doors is called “acceptable wind resistance door” that is constrained by the fact that the “Crb\_Measure\_Pressure\_kn\_m2” quantity has a minimum value of 0,8 (this explicit subclass is not visible in the UML diagram).

As an example: in the EXPRESS variant for the EN ISO 12006-3 value constraint this means:

```
ENTITY xtdValue
  SUBTYPE OF (xtdObject);
    NominalValue : OPTIONAL STRING;
    UpperTolerance : OPTIONAL STRING;
    LowerTolerance : OPTIONAL STRING;
    ValueType : xtdDataTypeEnum;
    ValueRole : OPTIONAL xtdValueRoleEnum;
    ToleranceType : OPTIONAL xtdToleranceTypeEnum;
    Order : INTEGER;
END_ENTITY;
```

- NominalValue gets the value “0,8”;
- ValueRole gets the value “MINIMUM”;
- ValueType gets the value “XTD\_REAL”; and
- Order gets the value “0”.

Furthermore, there is an “access control system (ACS)” controlling multiple doors for which a attribute called “number of connections” is stored.

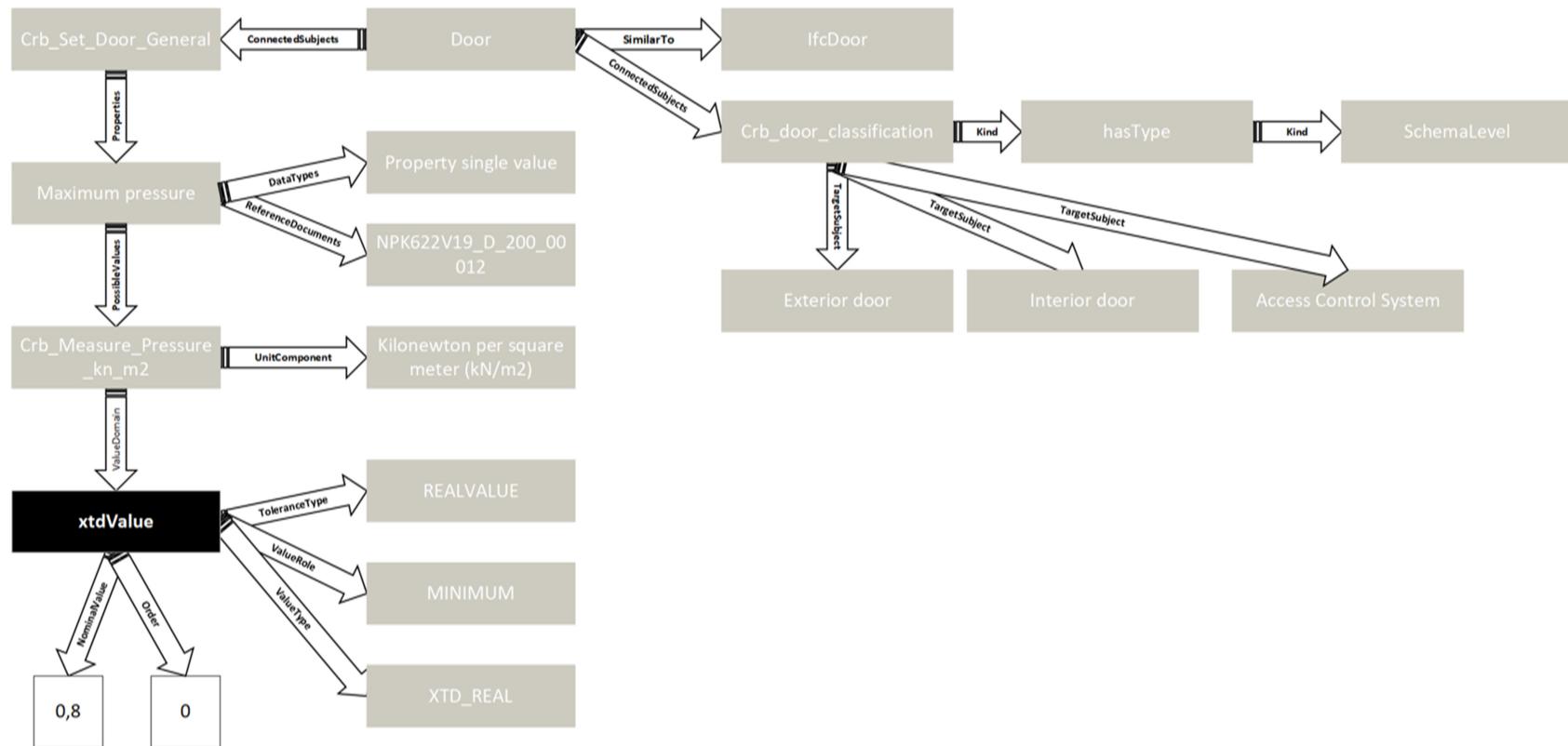


Figure D.4 — CRB ontology fragment (schematically)

The example is now extended in two ways (Figure D.5):

- A boolean attribute “isExternal” is added including a constraint for exterior door (being true); and
  - It is indicated that an Exterior door always has a specific CRB classification code (eBKP\_H = E 3.2).
- Not in the diagram: an ACS always has another CRB classification code (eBKP\_H = D 3.2).

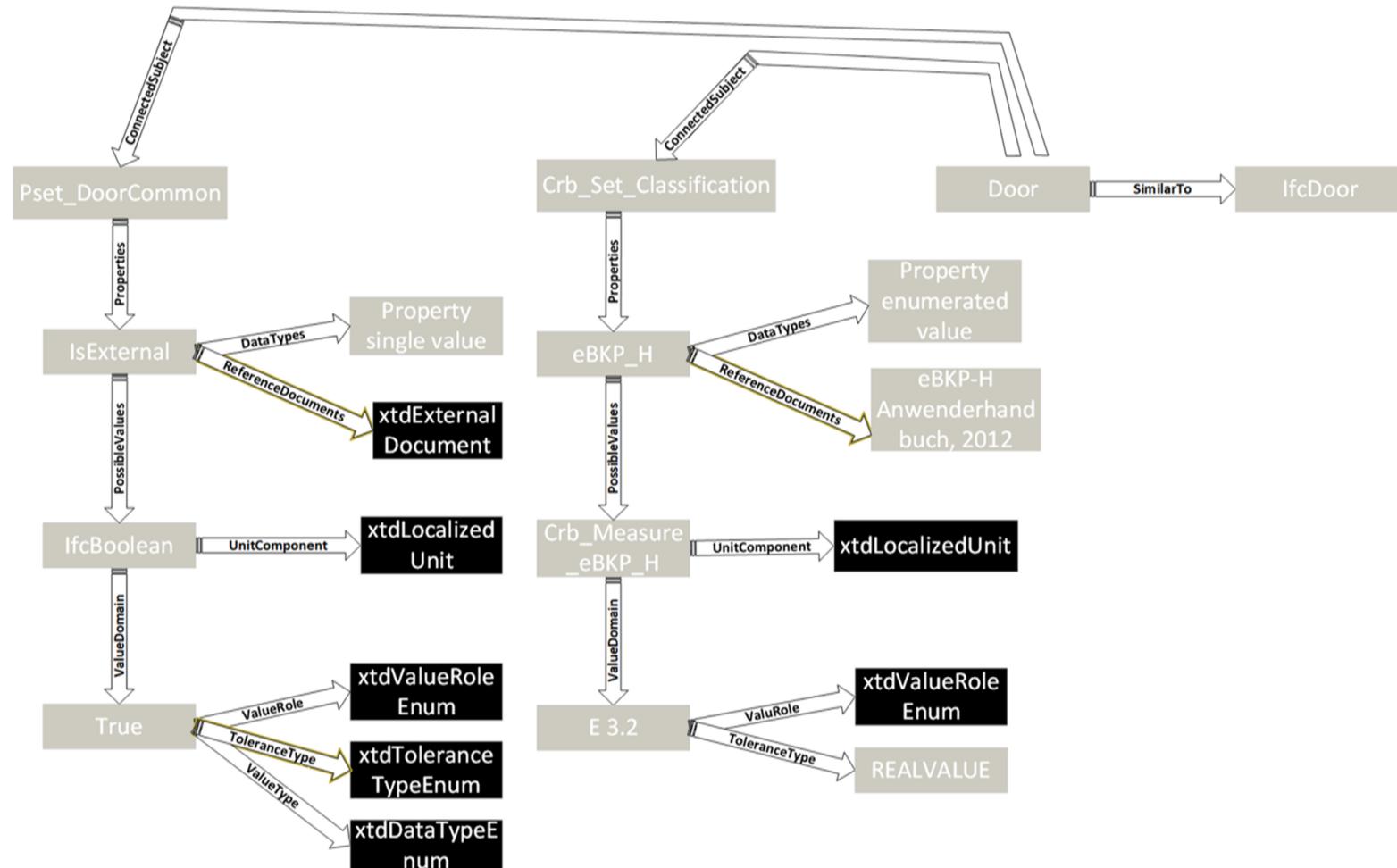
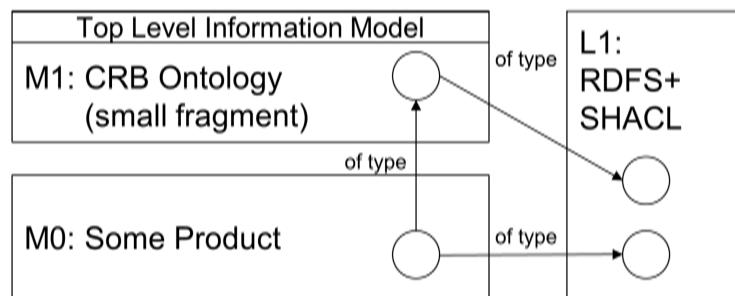


Figure D.5 — Extending the example

For machine-interpretability, the diagrams given above should (in the end) be represented in EXPRESS or XSD (not in this example).

This example is now mapped according to SML. The “LoC-3 RDFS+SHACL” variant is selected as L1: information language to e.g. be able to explicitly model the relevant minimum value in a CWA-fashion so that product instantiations can be automatically verified. As data format Turtle is used. This situation is pictured in Figure D.6.



**Figure D.6 — Mapped to SML**

#### An ‘empty’ crb.ttl ontology is defined first:

```

# baseURI: https://w3id.org/crb/def
# imports: https://w3id.org/sml/shacl/def
# prefix: crb

@prefix crb: <https://w3id.org/crb/def#> .
@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 sh: <http://www.w3.org/ns/shacl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sml: <https://w3id.org/sml/def#> .
@prefix unit: <http://qudt.org/vocab/unit> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/crb/def>
    a owl:Ontology ;
    owl:imports <https://w3id.org/sml/shacl/def> ;

```

In this ontology a “door” becomes a subclass of the predefined top level sml:PhysicalObject class. Additionally (not in UML), multi-lingual labels were added.

```

crb:Door
    a rdfs:Class ;
    rdfs:subClassOf sml:PhysicalObject ;
    skos:prefLabel "Deur"@nl ;
    skos:prefLabel "Door"@en ;
    skos:prefLabel "Porta"@it ;
    skos:prefLabel "Porte"@ch-fr ;
    skos:prefLabel "Porte"@fr ;

```

**SS-EN 17632-1:2022 (E)**

```
skos:prefLabel "Tür"@ch-de ;
skos:prefLabel "Tür"@de ;
```

Next, it is subclassed to an “interior door” and an “exterior door”, again adding example multi-lingual labels (some even showing language regions):

```
crb:InteriorDoor
a rdfs:Class ;
rdfs:subClassOf crb:Door ;
skos:prefLabel "Interior door"@en ;
skos:prefLabel "Innentür"@ch-de ;
skos:prefLabel "Porte intérieure"@ch-fr ;
```

```
crb:ExteriorDoor
a rdfs:Class ;
rdfs:subClassOf crb:Door ;
skos:prefLabel "Aussentür"@ch-de ;
skos:prefLabel "Außentür"@de ;
skos:prefLabel "Buitendeur"@nl ;
skos:prefLabel "Exterior door"@en ;
skos:prefLabel "Porta esterna"@it ;
skos:prefLabel "Porte extérieure"@ch-fr ;
skos:prefLabel "Porte extérieure"@fr ;
```

The “exterior door” is further subclassed into a:

```
crb:AcceptableMechanicalPressureLoadDoor
a rdfs:Class ;
rdfs:subClassOf crb:ExteriorDoor ;
```

Next, the pressure property is defined (no need to explicitly put the unit in the name) together with a link to a related reference document on the web.

```
crb:maxPressure
a rdf:Property ;
sml:unit unit:KiloPA ;
rdfs:comment "D.200.00012 (NPK 622, 2019)" ;
rdfs:seeAlso <https://npkviewer.crb.ch> ;
```

NOTE 1 The unit is defined on property definition level (for all instances) as in EN ISO 12006-3.

Next, the boolean isExternal is defined:

```
crb:isExternal
a rdf:Property ;
rdfs:range xsd:Boolean ;
```

Next, the classification code property is defined together with a subset of the allowed instances (the ones used here in the example):

```
crb:eBKP-HType
a rdf:Property ;
rdfs:range crb:EBKP-HType ;
```

```
crb:EBKP-HType
a rdfs:Class ;
```

```

a sml:EnumerationType ;
.

crb:D32
a crb:EBKP-HType ;
.

crb:E32
a crb:EBKP-HType ;
.

crb:G14
a crb:EBKP-HType ;
.

```

NOTE 2 The dots and spaces in the original EBKP-HType value enumerations (and therefore the relevant instances) are deleted (not allowed by RDF in such a name). If needed they can be reintroduced via adding skos:prefLabels.

Next, an attribute of doors is defined that is needed for a quantity analysis for door construction costs evaluations. One for interior doors and one for exterior doors.

```

crb:interiorDoorSurfaceArea
a rdf:Property ;
skos:prefLabel "Interior door surface area"@en ;
skos:prefLabel "Fläche Innentür, Innentor"@ch-de ;
skos:prefLabel "Surface de porte intérieure"@ch-fr ;
sml:abbreviation "FII"@ch-de ;
sml:abbreviation "SPINT"@ch-fr ;
sml:unit unit:M2 ;
rdfs:comment "eBKP-H Anwenderhandbuch, 2012" ;
rdfs:seeAlso <https://www.crb.ch/Normen-Standards/Baukostenplaene/eBKP\_H.html> ;
.

```

and

```

crb:exteriorDoorSurfaceArea
a rdf:Property ;
skos:prefLabel "Exterior door surface area"@en ;
skos:prefLabel "Fläche Tür, Tor"@ch-de ;
skos:prefLabel "Surface de porte"@ch-fr ;
sml:abbreviation "FTT"@ch-de ;
sml:abbreviation "SPOR"@ch-fr ;
sml:unit unit:M2 ;
rdfs:comment "eBKP-H Anwenderhandbuch, 2012" ;
rdfs:seeAlso <https://www.crb.ch/Normen-Standards/Baukostenplaene/eBKP\_H.html> ;
.

```

Next, the “access control system” (ACS) is defined by:

```

crb:AccessControlSystem
a rdfs:Class ;
rdfs:subClassOf sml:PhysicalObject ;
skos:definition "Included in the element are the systems for the automatic checking of access authorization, the control of locks and the

```

**SS-EN 17632-1:2022 (E)**

registration of processes (entry and exit control systems, handling systems in parking garages, badge readers, light barriers and the like)"@en ;

```
skos:prefLabel "Access control system"@en ;
skos:prefLabel "Installations de contrôle d'accès"@ch-fr ;
skos:prefLabel "Zutrittskontrollanlage"@ch-de ;
sml:abbreviation "ACS"@en ;
```

Having the property:

```
crb:accessControlSystemConnectionPointCount
a rdf:Property ;
rdfs:comment "eBKP-H Anwenderhandbuch, 2012" ;
rdfs:seeAlso <https://www.crb.ch/Normen-Standards/Baukostenplaene/eBKP\_H.html> ;
skos:prefLabel "Access control system connection point count"@en ;
skos:prefLabel "Anzahl anschlusspunkte zutrittskontrollanlage"@ch-de ;
skos:prefLabel "Nombre de points de raccordement"@ch-fr ;
sml:abbreviation "AAZK"@ch-de ;
sml:abbreviation "NPRIC"@ch-fr ;
```

The following SHACL shape is added for “Door”:

```
crb:DoorShape
rdf:type sh:NodeShape ;
sh:property [
    a sh:PropertyShape ;
    sh:path crb:eBKP-HType ;
    sh:class crb:EBKP-HType ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
]
sh:property [
    rdf:type sh:PropertyShape ;
    sh:path crb:isControlledBy ;
    sh:class crb:AccessControlSystem ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
]
sh:targetClass crb:Door ;
```

The following SHACL shapes are added specifically for “interior door” respectively “exterior door”:

```
crb:InteriorDoorShape
a sh:NodeShape ;
sh:property [
    rdf:type sh:PropertyShape ;
    sh:path (
        crb:interiorDoorSurfaceArea
```

```

        rdf:value
    ) ;
    sh:datatype xsd:double ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
] ;
sh:property [
    a sh:PropertyShape ;
    sh:path crb:eBKP-HType ;
    sh:class crb:EBKP-HType ;
    sh:hasValue crb:G14 ;
] ;
sh:property [
    a sh:PropertyShape ;
    sh:path crb:isExternal ;
    sh:datatype xsd:boolean ;
    sh:hasValue false ;
] ;
sh:targetClass crb:InteriorDoor ;
.

crb:ExteriorDoorShape
a sh:NodeShape ;
sh:property [
    rdf:type sh:PropertyShape ;
    sh:path (
        crb:exteriorDoorSurfaceArea
        rdf:value
    ) ;
    sh:datatype xsd:double ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
] ;
sh:property [
    a sh:PropertyShape ;
    sh:path crb:eBKP-HType ;
    sh:hasValue crb:E32 ;
] ;
sh:property [
    a sh:PropertyShape ;
    sh:path crb:isExternal ;
    sh:hasValue true ;
] ;
sh:targetClass crb:ExteriorDoor ;
.

```

The following SHACL shape is added specifically for “acceptable wind resistance door”:

```

crb:AcceptableMechanicalPressureLoadDoorShape
a sh:NodeShape ;
sh:property [
    a sh:PropertyShape ;
    sh:path (
        crb:maxPressure
        rdf:value

```

**SS-EN 17632-1:2022 (E)**

```

        ) ;
    sh:datatype xsd:double ;
    sh:minInclusive "0.8"^^xsd:double ;
]
sh:targetClass crb:AcceptableMechanicalPressureLoadDoor ;
.
```

Also for the AccessControlSystem a shape is defined:

```

crb:AccessControlSystemShape
a sh:NodeShape ;
sh:property [
    a sh:PropertyShape ;
    sh:path crb:eBKP_HType ;
    sh:class crb:EBKP_HType ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:hasValue crb:D32 ;
]
sh:property [
    a sh:PropertyShape ;
    sh:path crb:accessControlSystemConnectionPointCount ;
    sh:datatype xsd:integer ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
]
sh:targetClass crb:AccessControlSystem ;
.
```

Finally a relation is defined between a Door and an ACS:

```

crb:isControlledBy
a rdf:Property ;
.
```

Unlike EN ISO 12006-3, this ontology can now be further instantiated using the same LD/SW approach in actual product instance information:

```

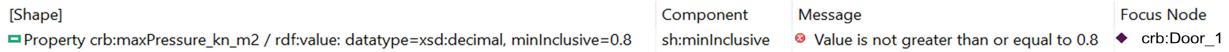
crb:Door_1
rdf:type crb:AcceptableMechanicalPressureLoadDoor ;
crb:eBKP-HType crb:E32 ;
crb:exteriorDoorSurfaceArea [
    rdf:value "2.4"^^xsd:double ;
]
crb:isControlledBy crb:ACS_1 ;
crb:isExternal true ;
crb:maxPressure [
    rdf:value "0.5"^^xsd:double ;
]
.

crb:ACS_1
a crb:AccessControlSystem ;
```

```

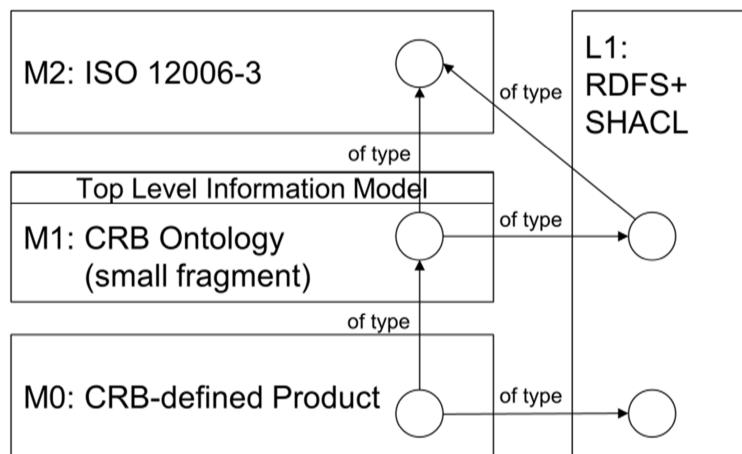
crb:accessControlSystemConnectionPointCount 5 ;
crb:eBKP-HType crb:D32 ;
.
.
```

This information can be automatically verified given these specific instances (Figure D.7).



**Figure D.7 — Automatic SHACL-based data verification**

In the future, when EN ISO 12006-3 would also be available in a 'linked data' language and format (say SHACL and Turtle) the two approaches (EN ISO 12006-3 and SML) could be combined as depicted in Figure D.8.



**Figure D.8 — Future combination potential**

This way, elements in the CRB Ontology will be typed towards their EN ISO 12006-3 model type and towards their LD language types.

```

crb:ExteriorDoor
  a rdfs:Class ; -- L1
  a sh:NodeShape ; -- L1
  a 12006-3:Subject ; -- M2
  rdfs:subClassOf crb:Door ;
  ...
.
```

#### D.3.4 Full CRB code example in RDFS/SHACL (in Turtle format)

```

# baseURI: https://w3id.org/crb/def
# imports: https://w3id.org/sml/shacl/def
# prefix: crb

@prefix crb: <https://w3id.org/crb/def#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .
```

**SS-EN 17632-1:2022 (E)**

```

@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 sh: <http://www.w3.org/ns/shacl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sml: <https://w3id.org/sml/def#> .
@prefix unit: <http://qudt.org/vocab/unit> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/crb/def>
  a owl:Ontology ;
  owl:imports <https://w3id.org/sml/shacl/def> ;

  crb:ACS_1
    a crb:AccessControlSystem ;
    crb:accessControlSystemConnectionPointCount 5 ;
    crb:eBKP-HType crb:D32 ;

  crb:AcceptableMechanicalPressureLoadDoor
    a rdfs:Class ;
    rdfs:subClassOf crb:ExteriorDoor ;

  crb:AcceptableMechanicalPressureLoadDoorShape
    a sh:NodeShape ;
    sh:property [
      a sh:PropertyShape ;
      sh:path (
        crb:maxPressure
        rdf:value
      ) ;
      sh:datatype xsd:double ;
      sh:minInclusive "0.8"^^xsd:double ;
    ] ;
    sh:targetClass crb:AcceptableMechanicalPressureLoadDoor ;

  crb:AccessControlSystem
    a rdfs:Class ;
    rdfs:subClassOf sml:PhysicalObject ;
    skos:definition "Included in the element are the systems for the automatic checking of access authorization, the control of locks and the registration of processes (entry and exit control systems, handling systems in parking garages, badge readers, light barriers and the like)"@en ;
    skos:prefLabel "Access control system"@en ;
    skos:prefLabel "Installations de contrôle d'accès"@ch-fr ;
    skos:prefLabel "Zutrittskontrollanlage"@ch-de ;
    sml:abbreviation "ACS"@en ;

  crb:AccessControlSystemShape
    a sh:NodeShape ;
    sh:property [
      a sh:PropertyShape ;
      sh:path crb:accessControlSystemConnectionPointCount ;
      sh:datatype xsd:integer ;
      sh:minCount 1 ;
    ]
  
```

```

        sh:maxCount 1 ;
    ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path crb:eBKP-HType ;
    sh:class crb:EBKP-HType ;
    sh:hasValue crb:D32 ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
] ;
sh:targetClass crb:AccessControlSystem ;

.
crb:D32
    a crb:EBKP-HType ;
.

crb:Door
    a rdfs:Class ;
    rdfs:subClassOf sml:PhysicalObject ;
    skos:prefLabel "Deur"@nl ;
    skos:prefLabel "Door"@en ;
    skos:prefLabel "Porta"@it ;
    skos:prefLabel "Porte"@ch-fr ;
    skos:prefLabel "Porte"@fr ;
    skos:prefLabel "Tür"@ch-de ;
    skos:prefLabel "Tür"@de ;
.

crb:DoorShape
    a sh:NodeShape ;
    sh:property [
        a sh:PropertyShape ;
        sh:path crb:eBKP-HType ;
        sh:class crb:EBKP-HType ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
    ] ;
    sh:property [
        a sh:PropertyShape ;
        sh:path crb:isControlledBy ;
        sh:class crb:AccessControlSystem ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
    ] ;
    sh:targetClass crb:Door ;
.

crb:Door_1
    a crb:AcceptableMechanicalPressureLoadDoor ;
    crb:eBKP-HType crb:E32 ;
    crb:exteriorDoorSurfaceArea [
        rdf:value "2.4"^^xsd:double ;
    ] ;
    crb:isControlledBy crb:ACS_1 ;
    crb:isExternal true ;
    crb:maxPressure [
        rdf:value "0.5"^^xsd:double ;
    ] ;

```

**SS-EN 17632-1:2022 (E)**

```

.
crb:E32
  a crb:EBKP-HType ;
.

crb:EBKP-HType
  a rdfs:Class ;
  rdfs:subClassOf sml:EnumerationType ;
.

crb:ExteriorDoor
  a rdfs:Class ;
  rdfs:subClassOf crb:Door ;
  skos:prefLabel "Aussentür"@ch-de ;
  skos:prefLabel "Außentür"@de ;
  skos:prefLabel "Buitendeur"@nl ;
  skos:prefLabel "Exterior door"@en ;
  skos:prefLabel "Porta esterna"@it ;
  skos:prefLabel "Porte extérieure"@ch-fr ;
  skos:prefLabel "Porte extérieure"@fr ;
.

crb:ExteriorDoorShape
  a sh:NodeShape ;
  sh:property [
    a sh:PropertyShape ;
    sh:path crb:eBKP-HType ;
    sh:class crb:EBKP-HType ;
    sh:hasValue crb:E32 ;
  ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path crb:isExternal ;
    sh:datatype xsd:boolean ;
    sh:hasValue true ;
  ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path (
      crb:exteriorDoorSurfaceArea
      rdf:value
    ) ;
    sh:datatype xsd:double ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
  ] ;
  sh:targetClass crb:ExteriorDoor ;
.

crb:G14
  a crb:EBKP-HType ;
.

crb:InteriorDoor
  a rdfs:Class ;
  rdfs:subClassOf crb:Door ;
  skos:prefLabel "Innentür"@ch-de ;
  skos:prefLabel "Interior door"@en ;
  skos:prefLabel "Porte intérieure"@ch-fr ;
.

```

```

crb:InteriorDoorShape
a sh:NodeShape ;
sh:property [
  a sh:PropertyShape ;
  sh:path crb:eBKP-HTYPE ;
  sh:class crb:EBKP-HTYPE ;
  sh:hasValue crb:G14 ;
]
sh:property [
  a sh:PropertyShape ;
  sh:path crb:isExternal ;
  sh:datatype xsd:boolean ;
  sh:hasValue false ;
]
sh:property [
  a sh:PropertyShape ;
  sh:path (
    crb:interiorDoorSurfaceArea
    rdf:value
  ) ;
  sh:datatype xsd:double ;
  sh:minCount 1 ;
  sh:maxCount 1 ;
]
sh:targetClass crb:InteriorDoor ;
.

crb:accessControlSystemConnectionPointCount
a rdf:Property ;
rdfs:comment "eBKP-H Anwenderhandbuch, 2012" ;
rdfs:seeAlso <https://www.crb.ch/Normen-Standards/Baukostenplaene/eBKP\_H.html> ;
skos:prefLabel "Access control system connection point count"@en ;
skos:prefLabel "Anzahl anschlusspunkte zutrittskontrollanlage"@ch-de ;
skos:prefLabel "Nombre de points de raccordement"@ch-fr ;
sml:abbreviation "AAZK"@ch-de ;
sml:abbreviation "NPRIC"@ch-fr ;
.

crb:eBKP-HTYPE
a rdf:Property ;
rdfs:range crb:EBKP-HTYPE ;
.

crb:exteriorDoorSurfaceArea
a rdf:Property ;
rdfs:comment "eBKP-H Anwenderhandbuch, 2012" ;
rdfs:seeAlso <https://www.crb.ch/Normen-Standards/Baukostenplaene/eBKP\_H.html> ;
skos:prefLabel "Exterior door surface area"@en ;
skos:prefLabel "Fläche Tür, Tor"@ch-de ;
skos:prefLabel "Surface de porte"@ch-fr ;
sml:abbreviation "FTT"@ch-de ;
sml:abbreviation "SPOR"@ch-fr ;
sml:unit unit:M2 ;
.

crb:interiorDoorSurfaceArea
a rdf:Property ;

```

## SS-EN 17632-1:2022 (E)

```

rdfs:comment "eBKP-H Anwenderhandbuch, 2012" ;
rdfs:seeAlso <https://www.crb.ch/Normen-Standards/Baukostenplaene/eBKP\_H.html> ;
skos:prefLabel "Fläche Innentür, Innentor"@ch-de ;
skos:prefLabel "Interior door surface area"@en ;
skos:prefLabel "Surface de porte intérieure"@ch-fr ;
sml:abbreviation "FII"@ch-de ;
sml:abbreviation "SPINT"@ch-fr ;
sml:unit unit:M2 ;

.
crb:isControlledBy
a rdf:Property ;

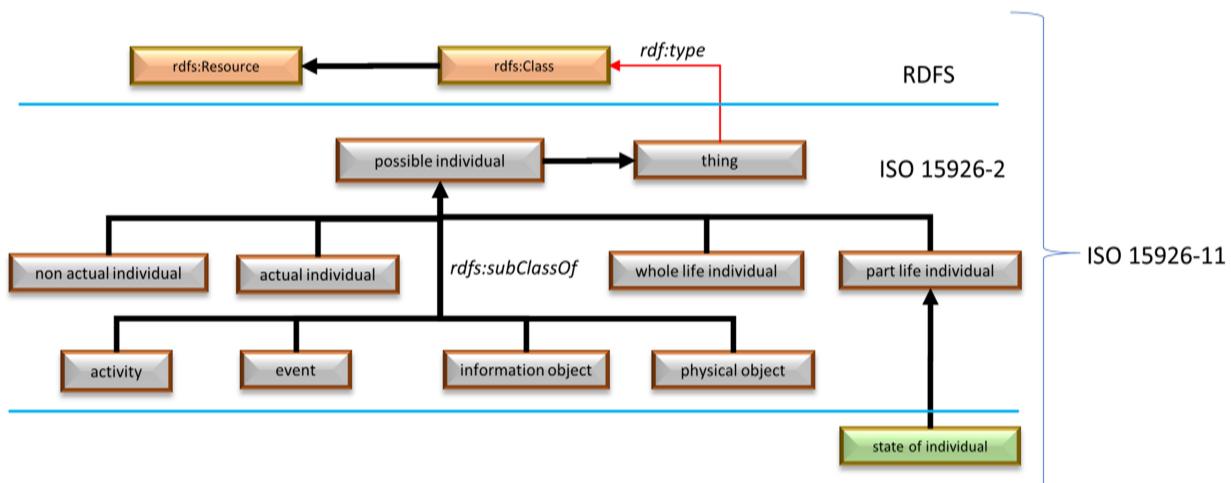
.
crb:isExternal
a rdf:Property ;
rdfs:range xsd:Boolean ;

.
crb:maxPressure
a rdf:Property ;
rdfs:comment "D.200.00012 (NPK 622, 2019)" ;
rdfs:seeAlso <https://npkviewer.crb.ch> ;
sml:unit unit:KiloPA ;
.
.
```

**D.4 Relationship with the ISO 15926 series**

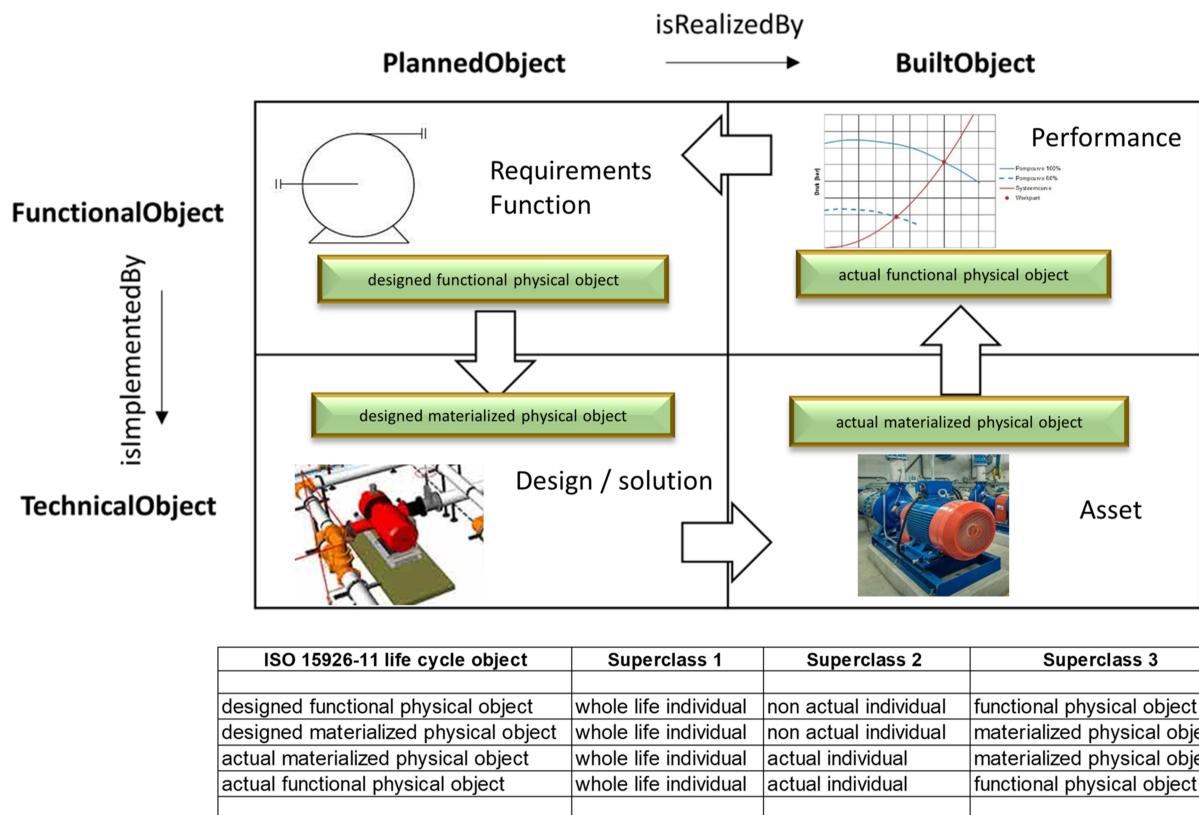
With this document, a bridge has been built between the construction and building industry and the process industry, where in the latter information integration has been under development for decades on the basis of, among other standards, the ISO 15926 series of standards. The ISO 15926 series also focuses on the W3C standards for a long time, with ISO/TS 15926-11 deliberately focusing on using only RDFS for the sake of simplicity.

Figure D.9 shows how the ISO/TS 15926-11 ed 2 standard in fact completely maps to the SML standard by connecting the information model of ISO 15926-2 to RDFS and expanding the latter with the entity “State of individual”.



**Figure D.9 — ISO 15926 series – SML information model correspondence**

Also the System Engineering principle of life cycle management covered by the SML standard is also part of the ISO 15926 series. In Figure D.10 it is shown how the quadrants of the life cycle model map to the entities of the ISO 15926 series.



**Figure D.10 — ISO 15926 series – SML SE correspondence**

## Annex E (informative)

### Linking information

#### **E.1 Types of linking**

The modelling of information models (here ontologies) and information sets is one thing. The (inter)linking of information models, the (inter)linking of information sets, and the linking of information sets to information models, is another. In the future, this linking will become increasingly important since information models and information sets are often interrelated in practice and complementing and strengthening each other in multidisciplinary decision making.

This linking can take place in various ways:

- Implicitly via, for example, a SPARQL query that questions multiple ontologies and/or information sets. The links are then implicitly encoded in the SPARQL code.
- Explicitly (and therefore more reusable) by declaratively modelling the links, preferably separately from the ontologies/information sets that are to be linked in 'link sets'. This way the link sets/links themselves remain 'agnostic' and 'clean'. This is also the way of linking elements of documents (information sets or information models) according to ISO 21597-1 (ICDD). Automatically inferring these links from the data, instead of asserting, is often a preferred approach requiring less governance.

Technically, from a 'linked data' technology perspective, a simple link set is a collection of RDF triples in which both the object and the subject of the triples can refer to elements from different ontologies or information sets. Every rdf:Property instance can be regarded as a "link"; so all relationships that are used in 'modelling' can be used for 'linking' too.

#### EXAMPLES

- disk:GirderBridge rdfs:subClassOf cbnl:Bridge;
- bdb:SomeBuilding owl:sameAs cdb:SomeBuilding;
- ex:Hospital\_1 sml:hasPart bim:Room-E3.64; and
- ex1:Asphalt\_123 ex2:canBeInspectedBy kvk:SomeCompany.

#### **E.2 Language-level language link sets**

The possibilities for linking ontologies and data sets depend to a large extent on level of capability and the corresponding 'linked data' language applied. Below it is specified which language-level constructs are typically used for different levels of capability.

##### **LoC-1 'weak' semantic linking**

- skos:exactMatch;
- skos:closeMatch;
- skos:narrowMatch;

- skos:broadMatch;
- skos:relatedMatch; and
- rdfs:seeAlso.

**LoC-2/LoC-3 'strong' semantic linking**

- rdfs:subClassOf between classes;
- rdfs:subPropertyOf between properties;
- via rdfs:domain or rdfs:range (i.e.: reference to a domain or range class/data type in an external ontology);
- + the "shorthands": owl:equivalentClass and owl:equivalentProperty; and
- owl:sameAs between individuals at information set level.

NOTE Information set level links are preferably avoided by classifying one individual with multiple rdf:type relationships to RDFS/OWL classes from different ontologies following the single source of facts (SSoF) principle. This approach is often not possible given multiple independent information sources governed by different information source holders who say something about the same phenomena in the real world.

**SS-EN 17632-1:2022 (E)**

## Bibliography

- [1] W3C Linked Data/Semantic Web (LD/SW):  
<https://www.w3.org/standards/semanticweb/data>
- [2] QUDT by qudt.org:  
<https://github.com/qudt/qudt-public-repo>
- [3] RFC 5646, Tags for Identifying Languages:  
<https://www.rfc-editor.org/rfc/bcp/bcp47.txt>
- [4] Cool URIs for the Semantic Web, W3C Interest Group Note 03 December 2008:  
<https://www.w3.org/TR/cooluris/#hashuri>
- [5] SHACL advanced features:  
<https://www.w3.org/TR/shacl-af/> (W3C track)  
<https://w3c.github.io/shacl/shacl-af/> (community track)
- [6] Defining N-ary Relations on the Semantic Web, W3C Working Group Note 12 April 2006:  
<https://www.w3.org/TR/swbp-n-aryRelations/>
- [7] EN ISO 19650-1, *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling - Part 1: Concepts and principles (ISO 19650-1)*
- [8] EN ISO 19650-4<sup>4</sup>, *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling - Part 4: Information exchange (ISO 19650-4)*
- [9] European Commission, Directorate-General for Informatics, New European interoperability framework: promoting seamless services and data flows for European public administrations, Publications Office, 2017:  
<https://data.europa.eu/doi/10.2799/360327>
- [10] EN ISO 80000 (all parts), *Quantities and units (ISO 80000 (all parts))*
- [11] EN ISO 16739-1:2020, *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries - Part 1: Data schema (ISO 16739-1:2018)*
- [12] OGC/W3C, Spatial Data on the Web Best Practices, W3C Working Group Note 28 September 2017:  
<https://www.w3.org/TR/sdw-bp/>
- [13] EN ISO 21597-1:2020, *Information container for linked document delivery - Exchange specification - Part 1: Container (ISO 21597-1:2020)*
- [14] EN ISO 21597-2:2020, *Information container for linked document delivery - Exchange specification - Part 2: Link types (ISO 21597-2:2020)*

---

<sup>4</sup> Under preparation. Stage at the time of publication: FprEN ISO 19650-4:2022.

- [15] EN ISO 23386:2020, *Building information modelling and other digital processes used in construction - Methodology to describe, author and maintain properties in interconnected data dictionaries (ISO 23386:2020)*
- [16] EN ISO 23387:2020, *Building information modelling (BIM) - Data templates for construction objects used in the life cycle of built assets - Concepts and principles (ISO 23387:2020)*
- [17] EN ISO 12006-3:2022, *Building construction - Organization of information about construction works - Part 3: Framework for object-oriented information (ISO 12006-3:2022)*
- [18] ISO/IEC/IEEE 12207:2017, *Systems and software engineering — Software life cycle processes*
- [19] ISO/IEC 11179-3:2013, *Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes*
- [20] EN ISO 9241-302:2008, *Ergonomics of human-system interaction - Part 302: Terminology for electronic visual displays (ISO 9241-302:2008)*
- [21] ISO/IEC 2382:2015, *Information technology — Vocabulary*
- [22] EN ISO 12006-2:2020, *Building construction - Organization of information about construction works - Part 2: Framework for classification (ISO 12006-2:2015)*
- [23] ISO 6707-3:2017, *Buildings and civil engineering works — Vocabulary — Part 3: Sustainability terms*
- [24] ISO/IEC 11179-3:2013, *Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes*
- [25] ISO 5127:2017, *Information and documentation — Foundation and vocabulary*
- [26] ISO 55000:2014, *Asset management — Overview, principles and terminology*

## Anteckningar/Notes

# Anteckningar/Notes



## Globala lösningar för ett smartare samhälle

SIS är en del av ISO och CEN som är nätverk av experter som arbetar med att skapa internationella standarder. Hos oss kan aktörer ta initiativ och samverka för best practice som främjar Sveriges konkurrenskraft och ger smart och hållbar samhällsutveckling. SIS samverkar med alla delar av det svenska samhället, som till exempel industri, akademi, offentlig sektor och frivilligorganisationer.

SIS projektleder det svenska arbetet med att ta fram standarder. Vi verkar för ökat svenskt inflytande i internationella samarbeten och för att best practice sprids och tillämpas i Sverige. Vi erbjuder också utbildningar, tjänster och produkter som hjälper våra kunder att utveckla sina verksamheter och skapa samhällsnytta med hjälp av standarder.

Vill du veta mer eller bidra till att skapa smarta globala lösningar, kontakta oss per telefon eller besök **sis.se**

Svenska institutet för standarder  
104 31 Stockholm  
Tel 08 - 555 523 10  
[kundservice@sis.se](mailto:kundservice@sis.se)  
[sis.se](http://sis.se)