CEN/TC 442

Date: 2020-11-11

prEN 17632:2021

Secretariat: SN

Building Information Modelling (BIM) –

Semantic Modelling and Linking (SML)

ICS:

Editorial Team:

Benno Koehorst (RWS, NL) – Chair

Michel Böhms (TNO, NL) – Editor

Sara Asmussen (LHB, GL)

Phil Jackson (University of Surrey, UK)

Davis Kantor (CRB, CH)

Ralf Klein (KU Leuven, BE)

Jacky Chi Ho Lau (bSI/buildingSMART Norway, NO)

Tim Lemoine (BBRI, BE)

Vlad Nyvlt (Institute of Technology and Business, CZ)

Giorgio Schiavini (DUEUNOSTUDIO, IT)

Espen Schulze (Cobuilder, NO)

Sanne Jansweijer (NEN, NL) – Secretary

**Contents** Page

[European foreword 4](#_Toc55830256)

[Introduction 5](#_Toc55830257)

[1 Scope 6](#_Toc55830258)

[2 Normative references 7](#_Toc55830259)

[3 Terms and definitions 8](#_Toc55830260)

[4 Symbols and abbreviated terms 12](#_Toc55830261)

[4.1 Symbols 12](#_Toc55830262)

[4.2 Abbreviated terms 12](#_Toc55830263)

[5 Levels of Capability 14](#_Toc55830264)

[6 Conceptual L1: Data language 15](#_Toc55830265)

[7 Concrete L1: Data language bindings 17](#_Toc55830266)

[7.1 Introduction 17](#_Toc55830267)

[7.2 Identification: URI strategy 21](#_Toc55830268)

[7.3 Naming conventions 22](#_Toc55830269)

[7.4 Annotation 22](#_Toc55830270)

[7.5 Enumeration datatypes 24](#_Toc55830271)

[7.6 Decomposition (instance level) 24](#_Toc55830272)

[7.7 Quantity kinds & units 24](#_Toc55830273)

[7.8 Quantity modelling 25](#_Toc55830274)

[7.9 Grouping 26](#_Toc55830275)

[8 Generic M1: Top level data model 26](#_Toc55830276)

[8.1 Top level model 26](#_Toc55830277)

[8.2 Systems engineering extension 29](#_Toc55830278)

[8.2.1 Introduction 29](#_Toc55830279)

[8.2.2 Conformance against requirements 30](#_Toc55830280)

[9 Implementing SML in code 31](#_Toc55830281)

[10 Linking data 31](#_Toc55830282)

[10.1 Types of linking 31](#_Toc55830283)

[10.2 Language-level language link sets 32](#_Toc55830284)

[11 Conformance 32](#_Toc55830285)

[11.1 General 32](#_Toc55830286)

[11.2 Conformance on language level 32](#_Toc55830287)

[11.3 Conformance on semantic level 33](#_Toc55830288)

[Annex A (normative) Conceptual data language and top level data model 34](#_Toc55830289)

[A.1 General 34](#_Toc55830290)

[A.2 in SKOS (Turtle format) 34](#_Toc55830291)

[A.3 in RDFS (Turtle format) 41](#_Toc55830292)

[A.4 in OWL (Turtle format) 51](#_Toc55830293)

[A.5 in SHACL (Turtle format) 64](#_Toc55830294)

[Annex B (normative) Selected W3C RDF language subsets 80](#_Toc55830295)

[B.1 General 80](#_Toc55830296)

[B.2 XML schema part 2: Datatypes 2nd edition 80](#_Toc55830297)

[B.3 Resource description framework (RDF) 80](#_Toc55830298)

[B.4 Simple knowledge organization system (SKOS) 81](#_Toc55830299)

[B.5 Resource description framework schema (RDFS) 81](#_Toc55830300)

[B.6 Web ontology language (OWL) 82](#_Toc55830301)

[B.7 Shape constraint language (SHACL) 83](#_Toc55830302)

[Annex C (informative) Example in SKOS 86](#_Toc55830303)

[C.1 General 86](#_Toc55830304)

[C.2 Example in SKOS (RDF/XML format) 86](#_Toc55830305)

[C.3 Example in SKOS (Turtle format) 87](#_Toc55830306)

[C.4 Example in SKOS (JSON-LD format) 89](#_Toc55830307)

[Annex D (informative) Example in RDFS 93](#_Toc55830308)

[D.1 General 93](#_Toc55830309)

[D.2 Example in RDFS (RDF/XML format) 93](#_Toc55830310)

[D.3 Example in RDFS (Turtle format) 96](#_Toc55830311)

[D.4 Example in RDFS (JSON-LD format) 100](#_Toc55830312)

[Annex E (informative) Example in OWL 106](#_Toc55830313)

[E.1 General 106](#_Toc55830314)

[E.2 Example in OWL (RDF/XML format) 106](#_Toc55830315)

[E.3 Example in OWL (Turtle format) 109](#_Toc55830316)

[E.4 Example in OWL (JSON-LD format) 113](#_Toc55830317)

[Annex F (informative) Example in SHACL 120](#_Toc55830318)

[F.1 General 120](#_Toc55830319)

[F.2 Example in SHACL (RDF/XML format) 120](#_Toc55830320)

[F.3 Example in SHACL (Turtle format) 123](#_Toc55830321)

[F.4 Example in SHACL (JSON-LD format) 128](#_Toc55830322)

[Annex G (informative) Relationships with other asset/product modelling standards 136](#_Toc55830323)

[G.1 General 136](#_Toc55830324)

[G.2 Relationship with ISO 21597 136](#_Toc55830325)

[G.3 Relationship with ISO 23387 136](#_Toc55830326)

[G.3.1 Introduction 136](#_Toc55830327)

[G.3.2 Modelling relationship with ISO 12006-3 137](#_Toc55830328)

[G.3.3 Example CRB 137](#_Toc55830329)

[G.3.4 Full CRB code example in SHACL (in Turtle format) 149](#_Toc55830330)

[Bibliography 156](#_Toc55830331)

# European foreword

This document (prEN 17632:2021) has been prepared by Technical Committee CEN/TC 422 “Building Information Modelling (BIM)”, the secretariat of which is held by SN - Norway.

This document is currently submitted to the CEN Enquiry.

# Introduction

The built environment is the context of this document. In the life cycle of buildings or infrastructure its assets need to be managed across their entire life cycle, involving programming, design, building and operation (as defined by ISO 19650 series), and the supply chain producing and delivering them. Vast amounts of valuable data about them are created, communicated in a diverse range of formats and data structures - and often lost again. In order to manage the assets efficiently and effectively according to the standards practised in asset management (as defined by ISO 55000 series), data needs to be findable, accessible, interoperable end reusable (FAIR)[[1]](#footnote-2)).

The world wide web consortium (W3C) provides so-called linked data (LD) and semantic web (SW) technologies [1] which are capable of giving data common form (syntax) and meaning (semantics), making data FAIR in a vendor neutral fashion.

The aim of this document is to standardize the application of this technology for the built environment in order to enable the data becoming FAIR. This document specifies how the construction and software industries apply this linked data and semantic web technology.

It hereby follows the principle to keep semantic modelling as simple and as standard as possible (Table 1).

Table 1 — Aiming for standard and simple semantics

|  |  |  |
| --- | --- | --- |
| Semantic | Standard | Proprietary |
| Simple | OK | IF NEEDED |
| Complex | IF NEEDED | NOT OK |

In others words, it is not the intention of this document to persuade anyone to shift the data structures they already have in place. On the contrary, it is the suggestion of this document to store, model, publish and link these data in a findable, accessible, interoperable and reusable manner (FAIR). To benefit the industry from planning and design to construction and operation.

This document complements other ISO standards without any overlap. In the Annex G, related ISO standards are listed and the exact relationship is described.

Application of this document to new or existing software will result in future proof, semantic data interoperability that is interoperability so that the meaning of the data model within the context of a subject area is understood by the participating systems [SOURCE: ISO/IEC 19941].

Furthermore, the data sets and data models become compatible, reusable, combinable and thereby integrally applicable. This document enables decision making in and over every life cycle phase and the supply chain involved.

Data management in the built environment is characterized by three main categories:

* Terms and definitions of data (on object level and attribute level);
* Data exchange (transfer of data from one party to another party) and data sharing (publishing of data by one party where it can be accessed by other parties);
* Data integration involving linking all data together.

For each of these categories the interoperability approaches apply, as defined by the enterprise interoperability framework (EIF) (ISO 11354-1):

1. *Unified approach*, featuring some *common meta-model*;
2. *Integrated approach*, featuring some *common forms;*
3. *Federated approach*, no common forms or meta-model but dynamic accommodation/adjustment.

These approaches, according to the EIF, are valid for business, process, service and data aspects.

# Scope

This document discusses an integrated and unified approach for data aspects, specifically for assets in the built environment, using EIF terminology.

The following data architecture (Figure 1) applies within each category.

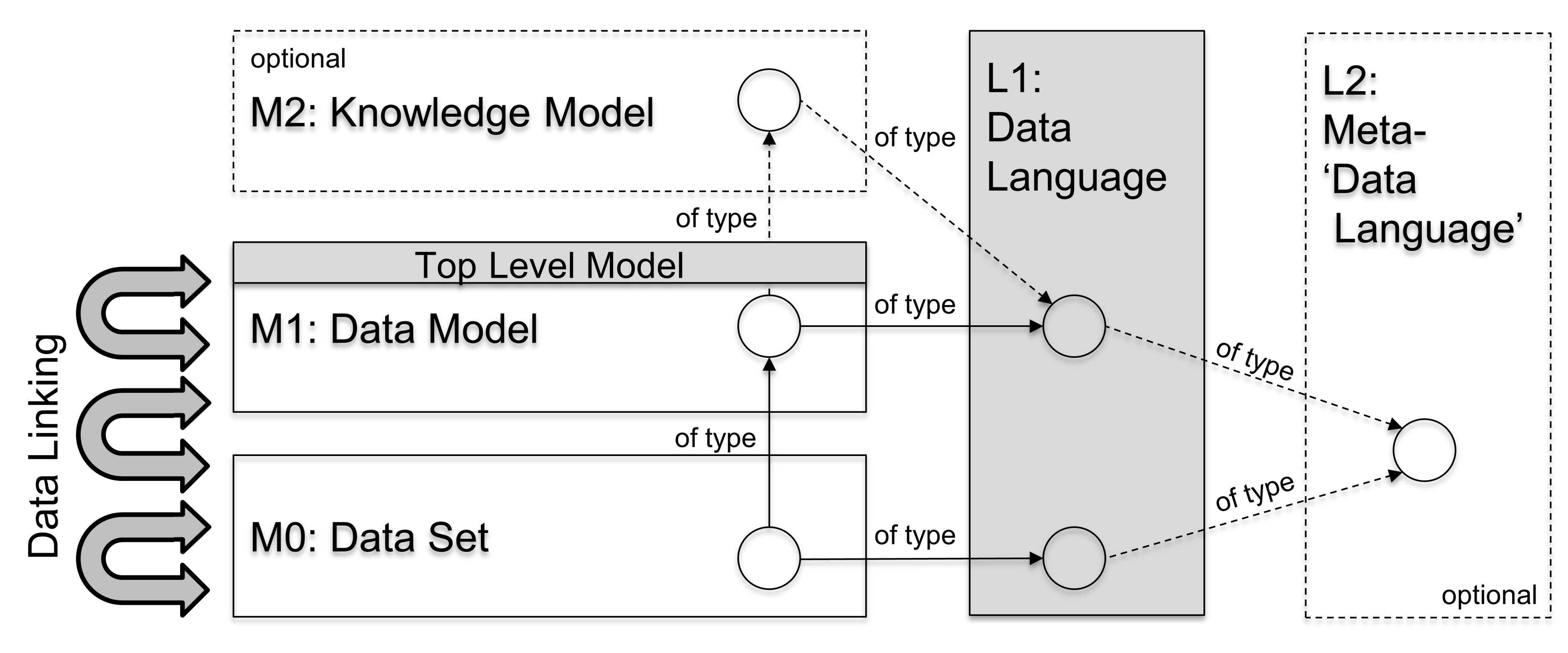


Figure 1 — Data architecture with typology (grey areas indicating the scope of this document)

This document specifies:

* a generic Top Level “M1: Data model” as common form;
* a conceptual “L1: Data language” as common meta-model with four ‘linked data’-based concrete language bindings (SKOS, RDFS, OWL and SHACL), including:
  + a choice of RDF-based formats (to be used for all modelling and language levels);
  + a set of data modelling patterns (for identification, naming, handling of enumeration types, quantity modelling, asset decomposition, grouping, etc.).
* a linking approach for interlinking data sets, interlinking data models and linking data sets and data models which are relevant within the built environment from many perspectives such as:
* Building information modelling (BIM);
* Geo-spatial information systems (GIS);
* Systems engineering (SE)[[2]](#footnote-3));
* Monitoring & control (M&C);
* Electronic document management (EDM).

This document does not specify a knowledge model since this is already available in ISO 12006-3.

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

The scope of this standard in general excludes the following:

* Business process modelling;
* Software implementation aspects;
* Data packaging and transportation/transaction aspects (handled by ISO TC59/SC13 Information container for document delivery (ICDD) respectively various information delivery manual (IDM) / information exchange requirements (EIR)-related initiatives);
* 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 upper ontology and modelling patterns.

# 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 Candidate Recommendation, 17 April 2020, <https://www.w3.org/TR/json-ld11/>

OWL 2[[3]](#footnote-4)) Web Ontology Language, Document Overview (Second Edition), W3C Recommendation, 11 December 2012, <https://www.w3.org/TR/2012/REC-owl2-overview-20121211/>

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

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

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

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

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

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

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

# 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 http://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

machine-readable

able to be read and processed by a computer

3.3

machine-interpretable

able to be semantically interpreted by a computer

3.4

level of capability

LoC

level of semantic level modelling power within a data model to fulfil a use case type

3.5

format

predetermined arrangement of data on a data medium

[SOURCE: ISO 5127:2017, 3.1.13.12]

3.6

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.

Note 2 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 2 to entry]

3.7

typology

hierarchy related to classification/instantiation relations

[SOURCE: ISO/IEC 11179-3:2013, 3.2.135, modified — The words "type of" have been removed. The words "which deals with" have been changed to "related to". The word relationships has been changed to relations.]

3.8

taxonomy

hierarchy related to generalization/specialization relations

[SOURCE: ISO/IEC 11179-3:2013, 3.2.135, modified — The words "type of" have been removed. The words "with deals with" have been changed to "related to". The word relationships has been changed to relations. Not mixing concepts, attributes and relations.]

3.9

meronomy

hierarchy related to part-whole (decomposition) relations

[SOURCE: ISO/IEC 11179-3:2013, 3.2.135, modified — The words "type of" have been removed. The words "with deals with" have been changed to "related to". The word relationships has been changed to relations.]

3.10

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]

3.11

triple

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

3.12

level of capability

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

3.13

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 draft, the terms instance and individual, are used as synonyms of object.

[SOURCE: ISO 12006-2:2015, 3.1.1, modified — added Note 2 to entry.]

3.14

concept

abstract entity for determining category membership

[SOURCE: ISO/IEC 2382 :2015, 2122971]

3.15

property

inherent or acquired feature of an object

3.16

attribute

data element for the computer-sensible description of a property, a relation or a class

[SOURCE ISO 22274:2013, 3.2]

3.17

relation

sense in which concepts can be connected, via constituent roles

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

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

3.18

data set

group of data instances directly specifying or describing something you can or could point at in reality

3.19

data model

specification/description of the organization of data giving meaning (semantics) to a data set

3.20

exchange information requirement

EIR

information requirement in relation to an appointment

[SOURCE: ISO 19650-1:2018. 3.3.6]

3.21

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]

3.22

metadata

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

3.23

top level data model

most generic taxonomy as part of a data model

# Symbols and abbreviated terms

## Symbols

This document does not contain any symbols.

## Abbreviated terms

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

|  |  |
| --- | --- |
| API | application programming interface |
| BIM | building information modelling |
| DT | data template [CEN TC 442] |
| ECMA | European computer manufacturers association international |
| EDM | electronic data management |
| EIF | enterprise interoperability framework |
| EIR | exchange information requirements |
| FAIR | findable, accessible, interoperable, reusable [go-fair.org] |
| FO | functional object |
| GIS | geo-spatial information systems |
| GUID | globally unique identifier (typically assigned) |
| ICDD | information container for linked document delivery [ISO] |
| ID | identifier |
| IDM | information delivery manual |
| IFC | industry foundation classes [ISO] |
| IETF | internet engineering task force |
| IO | imaginary object |
| JSON | JavaScript object notation [ECMA] |
| JSON-LD | JavaScript object notation - linked data [W3C] |
| LBD CG | linked building data community group [W3C] |
| LD | linked data technology [W3C] |
| LoC | level of capability |
| M&C | monitoring & control |
| OMG | object management group |
| OWL | web ontology language [W3C] |
| QUDT | quantities, units & data types [qudt.org] |
| RDF | resource description framework [W3C] |
| RDFS | resource description framework schema [W3C] |
| RFC | request for comments [IETF] |
| RO | real object |
| 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] |
| STEP | standard for the exchange of product model data [ISO] |
| SSoF | single source of facts |
| SW | semantic web technology [W3C] |
| TO | technical object |
| 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] |

# Levels of Capability

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

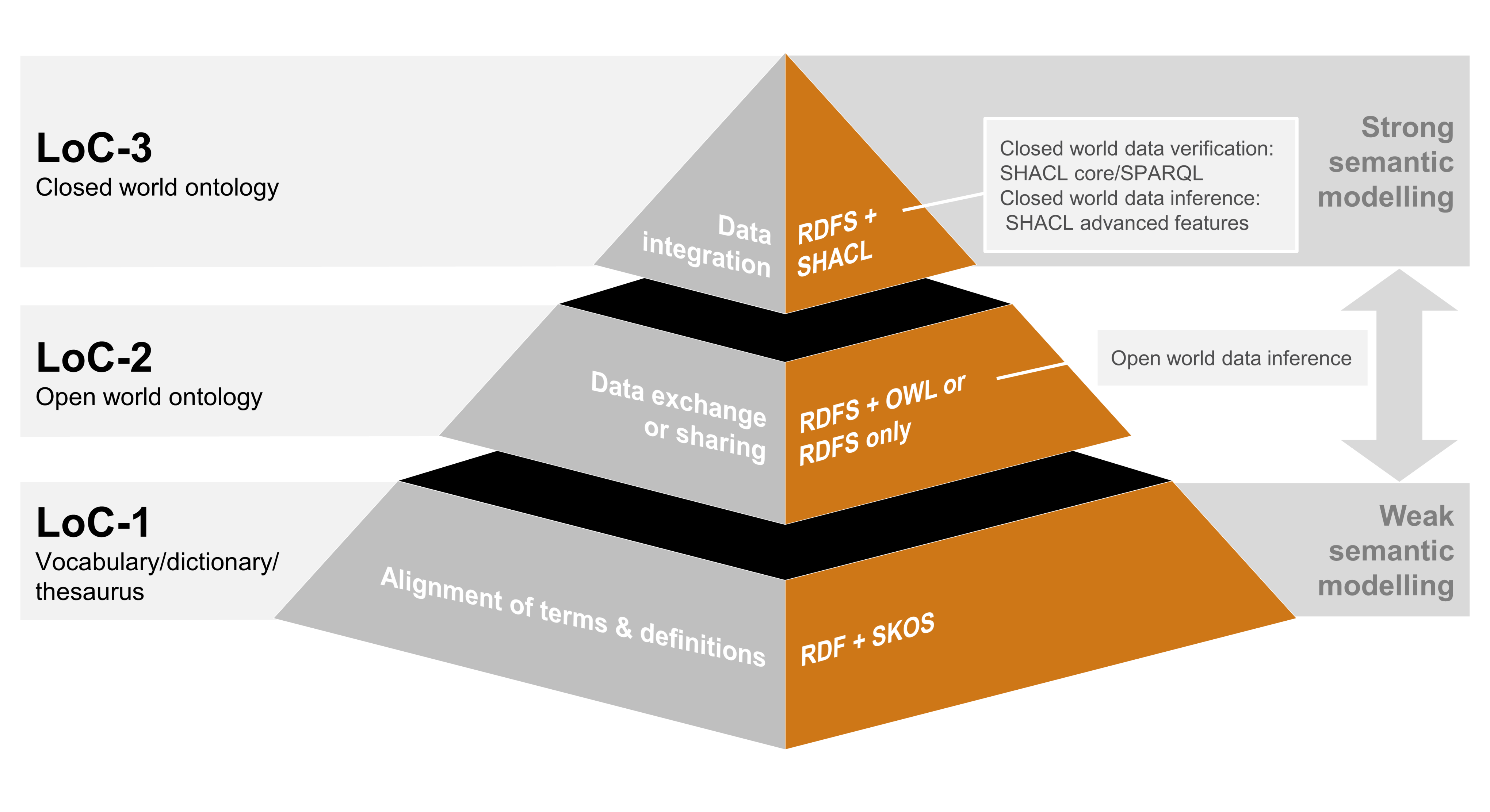


Figure 2 — Three main use case types and related 'Levels of Capability (LoCs)'

The left/grey part of the figure represents the business side, the right/orange side the modelling language used. 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 for human interpretation. A good definition gives an end user guidance on how to later classify and instantiate their data according to these terms. This level targets mainly uniformity in human understanding of terms and definitions and at least making sure the data is machine-*processable*, with the lowest level of capability or LoC-1. Terms and definitions can be distributed as well as published on websites for others to refer to and reuse. For this LoC-1 only RDF and the SKOS language shall be used.

Whenever uniformity is needed to *exchange or share* asset data between digital systems of different parties, more expressive power is needed in the form of LoC-2 where data is classified according to ontologies involving concepts, datatypes, attributes and relations or restrictions. This stronger level builds upon the lowest level making the data also machine-*interpretable*. Because of this adding of semantics, automatic (open world) inference of data from asserted data becomes possible. For this LoC-2 the RDFS or (RDFS + OWL) languages shall be used.

NOTE OWL is defined on top of RDFS and is providing more ‘logical’ modelling power than RDFS by introduction of ‘restrictions’ that can be used to infer new data from asserted data.

Finally, to be able to really *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 in a closed world fashion. 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 (RDFS+) SHACL (“core” for checking, “advanced features” for inference) language shall be used. For more info on SHACL advanced features let is not yet a recommendation (and hence not in the normative references) see [5].

# Conceptual L1: Data language

The ‘concrete language’-independent Conceptual L1: Data language is specified as in Table 2.

Table 2 — Conceptual L1: Data language

|  |  |  |
| --- | --- | --- |
| Meta-set | | |
| 1. Data model | | |
| 1. Data set | | |
| 1. Group | | |
| Meta-concept | | |
| 1. Concept | | |
| 1. Individual | | |
| 1. Value type | | |
| 1. Value | | |
| **Meta-relation** | ***Source*** | ***Target*** |
| 1. **Attribute** |  |  |
| 1. Annotation | 1 to 11 (or) | 7 |
| 1. Quality | 4 or 5 | 7 |
| 1. Quantity | 4 or 5 | 7 |
| 1. **Relation** |  |  |
| 1. Grouping | 3 to 11 (or) | 1 to 3 (or) |
| 5 | 1 to 3 (or) |
| 1. Classification (inverse: Instantiation) | 5 | 4 |
| 7 | 6 |
| 1. Generalisation (inverse: Specialisation) | 4 | 4 |
| 6 | 6 |
| 8 | 8 |
| 9 | 9 |
| 1. Composition (inverse: Decomposition) | 4 | 4 |
| 5 | 5 |
| 1. Association | 4 | 4 |
| 5 | 5 |
| 1. **Constraint** |  |  |
|  | 4 to 6  (or) | 8 or 9 |
|  | 8 or 9 | 4 to 6 (or) |
| 1. **Derivation** |  |  |
|  | 4 to 6 (or) | 8 or 9 |
|  | 8 or 9 | 4 to 6 (or) |
| 1. **Meta-role** | 4 or 5 | 9 |

This conceptual data language contains meta-sets, meta-concepts, meta-relations and a meta-role.

**Meta-sets**

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

**Meta-concepts**

The first basic meta-concept being a member of a data 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 datatype like a string, boolean or enumeration type that has finally values as instances.

**Meta-relations**

Two main forms of meta-relations are distinguished:

* Attributes
* Relations

Attributes describe intrinsic characteristics and relations describe extrinsic characteristics of individuals of concepts. Attributes are further divided in *annotations*, adding human-interpretable IDs, names, labels, definitions etc. and computer-interpretable *qualities* and *quantities*.

Three mechanisms as specific relations are defined:

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

Concepts shall be instantiated with Individuals referring to real world things you can (or ‘could’ in principle in case of an imaginary item like a plan or design) point at. Such instances get lexical values for attributes or references for relations. Lexical values shall be classified according to some value type (like string, decimal, integer, boolean etc.).

* Generalization (inverse: specialization), 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.

* Composition (inverse: decomposition), from ‘detailed’ to ‘global’.

Concepts shall be decomposed in concepts; Instances shall be decomposed in other instances.

These three mechanisms generate three hierarchy types namely:

* a *typology* (of concepts);
* a *taxonomy* (of concepts, attributes or relations);
* a *meronomy* (of concepts) respectively.

Beside the three abstraction mechanisms the following other relation types shall be used:

* Grouping, like for grouping all concepts, attributes, relations etc. into one data model;
* General associations on both concept and individual level;
* 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);

Derivations that tell us how new values for attributes or relations shall be inferred from existing asserted values.

Constraints shall be “definitional” or represent requirements, regulations or recommendations.

Associations in this document shall be in general n-ary associations. However, all specified language bindings in this document use binary associations only. In case there is a need to model n-ary relations (more specifically associations) the approach shall be used as described in [6].

**Meta-role**

A meta-role is only needed in case of n-ary relations and when no procedure is used to transform them to binary relations (like in [6]). Such a meta-role defines the involvement of a concept or individual in a relationship. In case of binary/directed relation such as in the linked data approach there is no need for such roles. The role coincides with the relation itself. This meta-aspect might however be useful for potential future non-LD language bindings like towards UML or the relational model.

# Concrete L1: Data language bindings

## Introduction

A language binding defines how the conceptual data language is mapped to the available modelling constructs provided by a given concrete data 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 3 and 4 the specifications are given for each modelling style per level of capability (LoC).

Table 3 — Specified language binding for LoC-1: Terminology and definition alignment with SKOS

|  |  |
| --- | --- |
| Meta- set/concept/relation |  |
| 1. Data model | skos:ConceptScheme |
| 2. Data set | not applicable |
| 3. Group | skos:Collection |
| 4. Concept | skos:Concept |
| 5. Individual | in general not applicable, only enumeration items: skos:Concept |
| 6. Value type | not applicable |
| 7. Value | not applicable |
| 8. Attribute |  |
| Annotation | existing skos annotations |
| Quality | skos:Concept |
| Quantity | skos:Concept |
| 9. Relation |  |
| Grouping | skos:inScheme, skos:member |
| Classification/ Instantiation | in general not applicable, only for enumeration items: skos:broader / skos:narrower |
| Generalisation/ Specialisation | skos:broader / skos:narrower |
| Composition/ Decomposition | skos:broader / skos:narrower |
| Association | skos:Concept |
| 10. Constraint | skos:related |
| 11. Derivation | skos:related |
| 12. Meta-role | n.a. |

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 class.

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

Table 4 — Specified language bindings for LoC-2&3: data exchange/sharing or integration

| Meta- set/concept/  relation | Language binding for LoC-2a: Data exchange/sharing with open world RDFS | Language binding for LoC-2b: Data exchange/sharing with open world (RDFS+) OWL | Language binding for LoC-3: Data integration with closed world (RDFS+) SHACL |
| --- | --- | --- | --- |
| 1. Data model | owl:Ontology | | |
| 2. Data set | owl:Ontology | | |
| 3. Group | rdfs:Container | | |
| 4. Concept | rdfs:Class | owl:Class | rdfs:Class |
| 5. Individual | rdfs:Resource (implicit) | owl:NamedIndividual or anonymous individual (implicit) | rdfs:Resource (implicit) |
| 6. Value type | 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 lists | rdfs:Datatype, or rdfs:Class + rdf:Property for enumeration data types |
| 7. Value | plain or typed literal (implicit) | plain or typed literal | plain or typed literal (implicit) |
| 8. Attribute |  |  |  |
| Annotation | rdf:Property | owl:AnnotationProperty | rdf:Property |
| Quality | rdf:Property  (except for enumerations) enumeration: rdf:Property with range a subclass of smls-rdfs:  EnumerationType class | owl:DatatypeProperty  (except for enumerations)  enumeration: owl:ObjectProperty with range a subclass of smls-owl:  EnumerationType class | rdf:Property  (except for enumerations)  enumeration: rdf:Property with sh:PropertyShape with sh:class subclass of smls-shacl:  EnumerationType |
| Quantity | rdf:Property with range smls-rdfs:  QuantityValue | owl:ObjectProperty with range smls-owl:  QuantityValue | rdf:Property with sh:PropertyShape with sh:class smls-shacl:QuantityValue |
| 9. Relation |  | | |
| Grouping | implicit (same file) or rdfs:member | | |
| Classification/ Instantiation | rdf:type | | |
| Generalisation/ Specialisation | rdfs:subClassOf or rdfs:subPropertyOf | rdfs:subClassOf & rdfs:subPropertyOf | rdfs:subClassOf & rdfs:subPropertyOf |
| Composition/ Decomposition | not available: to model via smls-rdfs:hasPart (instance level, constrained on class-level) | not available: to model via smls-owl:hasPart (instance level, constrained on class-level) | not available: to model  via smls-shacl:hasPart (instance level, constrained on class-level) |
| Association | rdf:Property | owl:ObjectProperty | rdf:Property |
| 10. Constraint | rdfs:domain or rdfs:range | owl:Restriction (+ details) rdfs:domain & rdfs:range | sh:NodeShape & sh:PropertyShape[[4]](#footnote-5)) |
| 11. Derivation | not applicable | not applicable | sh:Rule |
| 12. Meta-role | n.a. | n.a. | n.a. |

NOTE 3 Since all standard computer 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). Examples are given for RDF-XML, Turtle and JSON-LD (Annex C to F).

When mapping to a concrete language the following modelling patterns shall be applied.

## Identification: URI strategy

The URI strategy shall be specified as follows in the Table 5:

Table 5 — The specified URI has the following general form

|  |  |
| --- | --- |
| uri := https://{internet domain}/[path]/{reference}, where  reference :=  (defa/{data model name}[(/|#)({concept}|{attribute}|{relation}|{reference individual}]) |  (datb/{data set name}[(/|#]){individual}]) |  (docc/{document name}[document extension][(/|#){identifier}]) | |
| Meta-symbols | |
| xyz | A predefined string |
| := | defined by (“production rule”) |
| { … } | obligatory component (to be filled in) |
| [ … ] | optional component (to be filled in) |
| ( … ) | grouping |
| | | Logic exclusive OR operator |
| a to denote a data model, concept, attribute, relation or reference individual  b to denote a data set or an individual in the real world  c to denote a document giving information about a data model, data set, concept, attribute, relation or (reference) individual | |

EXAMPLE 1 http://www.interlink.eu/data/def/roadotl#Road

EXAMPLE 2 http://www.roadnetwork.nl/id/nwb#A16

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

NOTE 1 "data model" and "data set" are optional fragments because they are sometimes covered by the domain and/or path.

NOTE 2 The choice between "/" or "#" depends on the expected size of the data models / data sets. if big: use “/”, more background information in [4].

NOTE 3 If there is a need for versioning in the URI, then following the approach is specified (versionless and versioned variants where the versionless always refers to the latest version):

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

NOTE 4 The URIs are preferably not only for identification but also for localization (making data models, data sets and their elements “dereferenceable”).

NOTE 5 Do not use file extensions in names of data models and data sets. Use standard "server negotiation" to select the correct representation (including human-readable introductions in i.e. .html).

NOTE 6If necessary, use internet services that offer stable URIs such as w3id.org or purl.org.

## Naming conventions

There are two main ways to uniquely name/identify things in data sets or data models:

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

Both approaches have their pros and cons and are allowed by this document. When developing and learning ontologies human-friendly names help while in an operational phase with software tools hiding the details, codes are more stable (they stay stable when terms change) and language-neutral. Codes also make it possible to have one as label term referring to different concepts.

In case codes are used, human-friendly labels in the form of RDFS or SKOS labels shall be used too.

In case of codes at least unique preferred labels in the context of one ontology shall be used, since alternative labels are not unique.

When using names in data models:

* Concept or value type names always start with a capital (upper case) letter (“CamelCase”);
* Attribute or relation names always start with a non-capital (lower case) letter (“camelCase”);
* Names are in preferably in English and singular.

Example names in data 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”; or
* 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.

## Annotation

Textual definitions shall reuse the simple knowledge organization system (SKOS)

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

This gives a bit more specific meaning than the more generic rdfs:comment.

And by the following specification:

rdfs:isDefinedBy shall be 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 6.

Table 6 — Specified language binding for annotations

|  |  |
| --- | --- |
| Annotation | Language binding |
| primary label for | skos:prefLabel (deterministic) |
| synonym for | skos:altLabel |
| code for | skos:notation |
| remark of author | skos:editorialNote |
| scope | skos:scopeNote |
| comments | rdfs:comment |
| external information | rdfs:seeAlso |
| example application | skos:example |
| internally defined by | skos:definition |
| externally defined by | rdfs:isDefinedBy |
| abbreviated by | smls:abbreviation (a subproperty of skos:altLabel) |

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;
* 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”.

## Enumeration datatypes

Enumeration datatypes are specified as a concept so that allowed enumeration items can become individuals with multi-lingual labels and other meta-data like descriptions, definitions, etc. Open versus fixed enumeration datatypes are distinguished. The fixed ones are specified using the open world ‘owl:oneOf’ construct; having a closed world ‘sh:in’ counterpart.

The name of the enumeration datatype is a capital-starting variant of the relationship having this datatype as range. For clarity is advised to extend it with a “Type” fragment.

**EXAMPLE**

ex:loadLevel

a owl:ObjectProperty ;

rdfs:domain ex:Vehicle ;

rdfs:range ex:LoadLevelType ;

.

ex:LoadLevelType

a owl:Class ;

rdfs:subClassOf smls-owl:EnumerationType ;

owl:oneOf (

ex:Light

ex:Normal

ex:Heavy

) ;

.

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

## Decomposition (instance level)

This document specifies a non-transitive hasPart relation that is not necessarily inverse functional and having an implicit inverse relationship. 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.

## 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 ISO/IEC 80000 standard (system, names, definitions, symbols, etc.).

## Quantity modelling

There are many ways to model attributes (qualities or quantities). All have advantages and disadvantages, now or in the future. The most simple and direct modelling of an attribute in e.g. OWL is by an owl:DatatypeProperty. This way it is however hard to model meta-data like the often desired unit for a value in case of a quantity.

Therefore this document specifies quantities as relationships (like in OWL as an owl:ObjectProperty) with as range (rdfs:range) a predefined smls:QuantityValue class.

This quantity value shall have a value via rdf:value (with range xsd:decimal, recorded by means of a restriction at the quantity value class), possibly a unit via smls:unit of any other meta-attribute. When it comes to meta-quantities themselves, these are treated in the same way (i.e. a nested/recursive application). A typical example in given in Figure 3.

NOTE 1 Just like normal relationships, qualities are kept simple; in practice, there is less demand for meta-data about qualities / relationships (certainly no units).

NOTE 2 In addition to modelling meta-data about the instantiation/value of the quantity, also meta-data about the quantity definition itself is often needed. The price paid here is that properties for properties are required (resulting in OWL Full).

NOTE 3 Because quantity 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 smls:QuantityValue is not needed. By giving each attribute the range smls:QuantityValue, this typing can be inferred.

**EXAMPLE (graphically)**

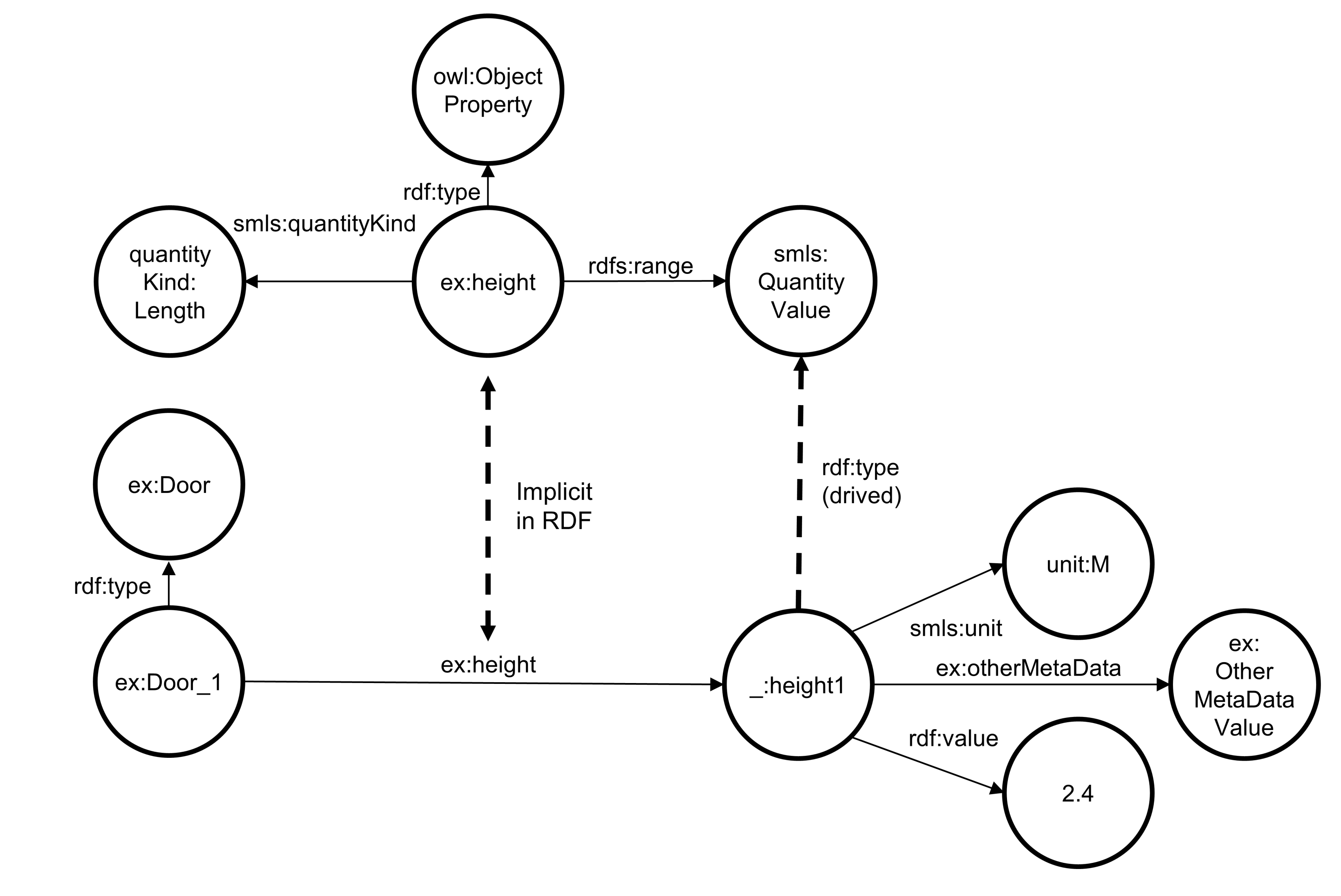


Figure 3 — Example quantity modelling

## Grouping

A grouping mechanism is added via rdfs:Container and rdfs:member. This standard mechanism shall be used, for example, to group attribute definitions, such as the "height", "width" and "depth" quantity definitions into "geometric quantities". Instance level grouping can this way also be derived.

# Generic M1: Top level data model

## Top level model

This document specifies the following top level concepts and their interrelations (actually: associations) and (hasPart) constraints:

* Entity
  + - Object
      * PhysicalObject
      * InformationObject
    - Activity
* State
* Event
* SpatialRegion
* TemporalRegion

A distinction is made in objects (subdivided into physical objects and information objects) that *exist* and activities that *happen*. The meta-model 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 one ‘snapshot’ of objects 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 are transformed by physical objects.

Entities, so objects and activities, live in spacetime and shall have a relevant interior and boundary in space and time (via spatial region respectively temporal region).

A spatial region shall be used for two purposes: defining the topological interior and the boundary of a physical object. For instance, by 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 specifies the following base modelling constructs to define a location in space and time:

* hasSpatialLocation;
* hasSpatialReferenceSystem;
* hasSpatialReference;
* SpatialReferenceSystem;
* SpatialReference;
* hasTemporalLocation;
* hasTemporalReferenceSystem;
* hasTemporalReference;
* TemporalReferenceSystem;
* TemporalReference.

Also some specific forms of spatial and temporal reference systems shall be predefined.

Finally, the top level model shall have ‘same-level’ relations (associations), restricted to their source and target concepts including class constraint variants of the composition/decomposition mechanism (also their inverses are given). They are specified in Table 7, their inverses in Table 8.

Table 7 — Specified top level relations

|  |  |  |
| --- | --- | --- |
| Associations | | |
| [Information­Object] | describes | [All] |
| [PhysicalObject] | hasState | [State] |
| [PhysicalObject] | triggers | [Event] |
| [PhysicalObject] | performs | [Activity] |
| [PhysicalObject] | hasPeriod | [TemporalRegion] |
| [PhysicalObject] | hasInterior | [SpatialRegion] |
| [PhysicalObject] | hasBoundary | [SpatialRegion] |
| [Activity] | hasState | [State] |
| [Activity] | transforms | [PhysicalObject] |
| [Activity] | transforms | [InformationObject] |
| [Activity] | triggers | [Event] |
| [Activity] | hasPeriod | [TemporalRegion] |
| [Activity] | hasInterior | [SpatialRegion] |
| [Activity] | hasBoundary | [SpatialRegion] |
| [Event] | begins | [State] |
| [Event] | ends | [State] |
| [State] | hasPeriod | [TemporalRegion] |
| [State] | hasInterior | [SpatialRegion] |
| [State] | hasBoundary | [SpatialRegion] |
| Composition/decomposition | | |
| [PhysicalObject] | hasPart | [PhysicalObject] |
| [InformationObject] | hasPart | [InformationObject] |
| [Activity] | hasPart | [Activity] |

Table 8 — Specified top level inverse relations

|  |  |  |
| --- | --- | --- |
| Inverse associations | | |
| [All] | isDescribedBy | [InformationObject] |
| [State] | isStateOf | [PhysicalObject] |
| [Event] | isTriggeredBy | [PhysicalObject] |
| [Activity] | isPerformedBy | [PhysicalObject] |
| [TemporalRegion] | isPeriodFor | [PhysicalObject] |
| [SpatialRegion] | isInteriorOf | [PhysicalObject] |
| [SpatialRegion] | isBoundaryOf | [PhysicalObject] |
| [State] | isStateOf | [Activity] |
| [PhysicalObject] | isTransformedBy | [Activity] |
| [InformationObject] | isTransformedBy | [Activity] |
| [Event | isTriggeredBy | [Activity] |
| [TemporalRegion] | isPeriodFor | [Activity] |
| [SpatialRegion] | isInteriorOf | [Activity] |
| [SpatialRegion] | isBoundaryOf | [Activity] |
| [State] | isBegunBy | [Event] |
| [State] | isEndedBy | [Event] |
| [TemporalRegion] | isPeriodFor | [State] |
| [SpatialRegion] | isInteriorOf | [State] |
| [SpatialRegion] | isBoundaryOf | [State] |
| Composition/decomposition | | |
| [PhysicalObject] | isPartOf | [PhysicalObject] |
| [InformationObject] | isPartOf | [InformationObject] |
| [Activity] | isPartOf | [Activity] |

When graphically combined, the top level (leaf) concepts and associations the model in Figure 4 is obtained.

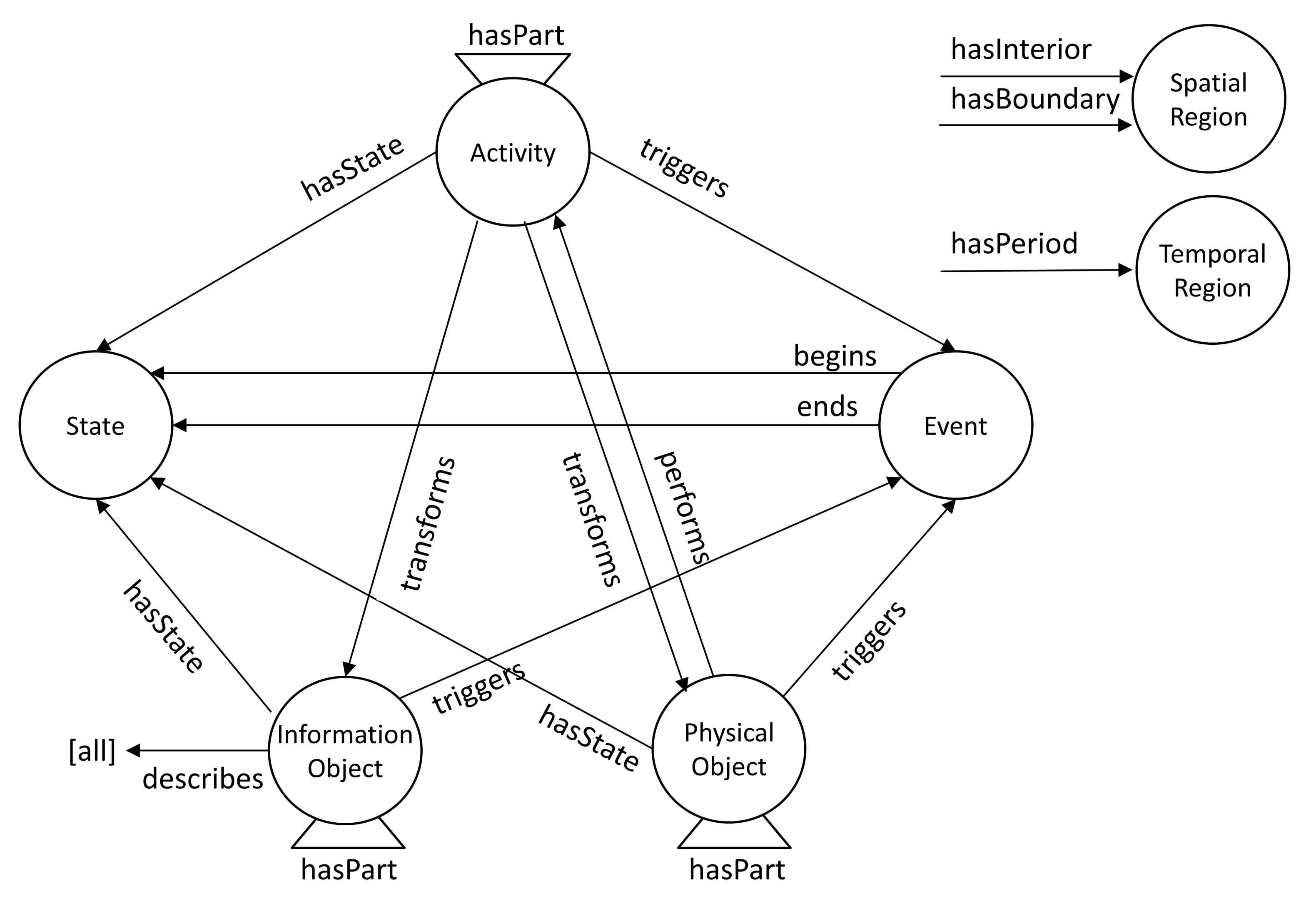


Figure 4 — Predefined concepts and relations combined

## Systems engineering extension

### Introduction

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

* Functional object versus technical object;
* Planned object versus built object.

NOTE 1 If an individual is classified as a kind this will normally hold for its whole lifetime. If something is a book sometime, it will normally always be a book. It is about what something “is”. If an individual is classified as a role 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 in time.

NOTE 2 That this dimension is completely orthogonal with “typical decomposition” (aka decomposition of class level): the distinction can be made on any scale level: networks, complexes, assets, element, 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 “FO-TO” jumps by subsequent parties in the supply chain.

When the two dimensions are combined (Figure 5), 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).

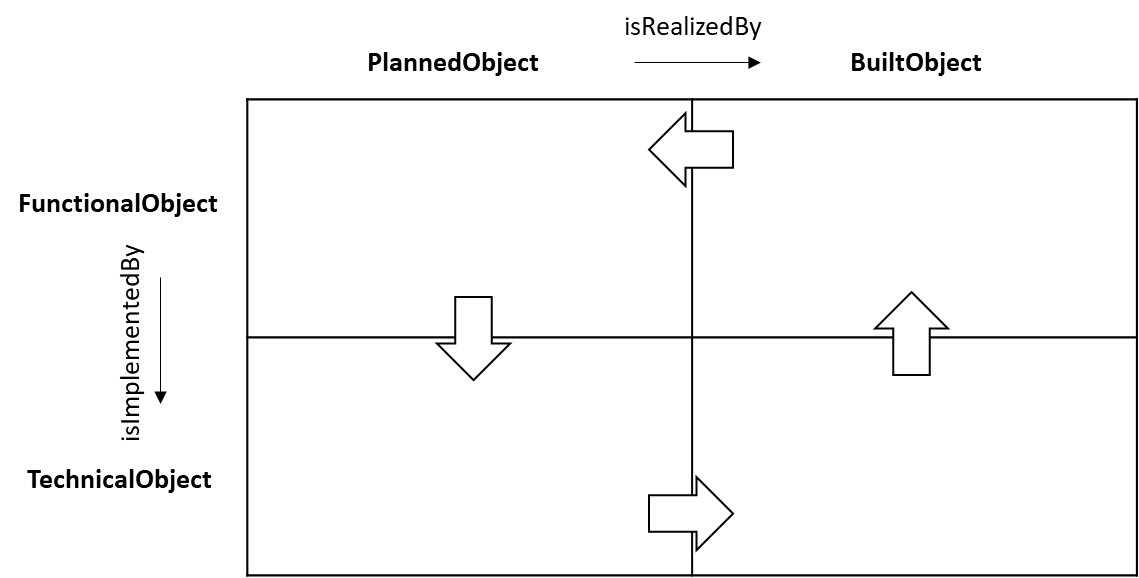


Figure 5 — Four quadrants relevant for systems engineering

A functional object shall perform a function being a subclass of our existing activity archetype. Like activity, this function always takes the form of a verb (not a noun) like “connecting”, “pumping”, “purchasing”, etc.

SML constructs summarized:

* smls:FunctionalObject;
* smls:TechnicalObject;
* smls:PlannedObject;
* smls:BuiltObject;
* smls:isImplementedBy (from smls:FunctionalObject to smls:TechnicalObject);
* smls:isRealizedBy (from smls:PlannedObject to smls:BuiltObject).

### Conformance against requirements

Functional objects (imaginary or real) shall have functional requirements (unstructured/textual and/or structured). Technical objects (imaginary or real) shall have technical requirements also known as ‘technical boundary conditions’ (again, unstructured/textual and/or structured)). Functional objects (imaginary or real) and technical objects (imaginary or real) can have observations. Observations for imaginary objects deal with planned, designed, calculated, simulated aspects. Observations for real objects deal with real aspects like measurements, tests, inspections etc.

The evaluation of requirements versus observations leads to levels of conformance. The check between requirements and imaginary observations (“planned performance”) is referred to as “verification”. The check between (verified) imaginary observations and real observations (“real performance”) is referred to as “validation”. Insufficient conformance leads to a decision-making process involving variants and preferred solutions based on multi-stakeholder/multi-criteria analysis resulting in a decision about the application of measures/activities being a technical (life cycle) activity: maintenance (kind of rebuild), renovation (involving redesign) etc. or a functional activity (like the reconsideration of the functional requirements).

A technical (life cycle) activity leads to a change in the state of a technical object that is being (re)designed and/or (re)build. A functional activity leads to a change in the state of a functional object that is being (re)purposed with the ultimate repurposing being ‘no more purpose’ aka demolishment (and potential recycling) of the object.

For the modelling of unstructured requirements this document specifies the following constructs:

* hasRequirement;
  + Requirement;
    - hasRequirementSpecification;
    - hasRequirementSource (a choice of: by definition, by client, by law or regulation or by recommendation);
    - hasRequirementsSeverity (a choice of: demand or wish).

# Implementing SML in code

All specifications above are implemented for the three LoCs in the four languages in the normative annex A. The format used is always Turtle. The four specified concrete language subsets are specified in the normative annex B.

NOTE An informative example is elaborated in Annexes C to F featuring all specified languages and formats.

# Linking data

## Types of linking

The modelling of data models (here ontologies) and data sets is one thing. The interlinking of data models, the interlinking of data sets, and the linking of data sets to data models, is another. In the future, this linking will become increasingly important since data models and data sets are often interrelated in practice and complementing each other in integrated decision making. The idea is to create networks or ecosystems of data and data structures that will form the basis for sector-wide digitization, integration and innovation as a digital system.

This linking can take place in various ways:

* Implicitly via, for example, a SPARQL query that questions multiple ontologies and/or data sets. The links are then implicitly encoded in the SPARQL code.
* Explicitly (and therefore better reusable) by declaratively modelling the links, preferably separately from the ontologies / data sets to be linked in ‘link sets’. This way the link sets/links themselves remain ‘agnostic’ and ‘clean’.

Technically, from a linked data technology perspective, a simple link set is a collection of RDF triples in which the object and subject of the triples refer to elements from different ontology or data set. 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

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

## 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 shall be used for each level of capability.

LoC-1 ‘weak’ semantic linking

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

LoC-2/3 ‘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;
* owl:sameAs between individuals at data level.

NOTE Data 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. Note that this is often not possible given multiple independent data sources governed by different data source holders who say something about the same phenomena in the real world.

# Conformance

## General

This clause defines two levels of conformance for data sets with respect to this SML standard.

## Conformance on language level

A data set is conformant on language level if, as prescribed by this document, it is:

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

NOTES

Example, equivalent, formats include RDF/XML, Turtle and JSON-LD.

Data models, reflect asset data from any perspective, including:

* Software-specific semantics;
* Person-specific semantics;
* Project-specific semantics;
* Department-specific semantics;
* Organization-specific semantics;
* Country-specific semantics;
* 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, at C2, data is both syntactically and semantically ‘liberated from the software’.

## Conformance on semantic level

A data set is conformant on semantic level if it reuses the generic top level data model (expressed in one of the standard RDF formats and using one of the standard linked data languages) as specified by this European norm.

C2 only makes sense on top of C1.

The actual conformance level required in practice can be specified in an exchange information requirement (EIR).

1. (normative)  
     
   Conceptual data language and top level data model
   1. General

This annex describes the SML standard implementation code in the selected linked data languages (SKOS, RDFS, OWL and SHACL) in the concrete syntax form Turtle. It covers both the defined conceptual data language and the common top level data model.

* 1. in SKOS (Turtle format)

# baseURI: https://w3id.org/def/smls-skos

# imports: http://www.w3.org/2004/02/skos/core

# prefix: smls-skos

@prefix smls-skos: <https://w3id.org/def/smls-skos#> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

<https://w3id.org/def/smls-skos>

a skos:ConceptScheme ;

owl:imports <http://www.w3.org/2004/02/skos/core> ;

skos:hasTopConcept smls-skos:Entity ;

skos:hasTopConcept smls-skos:State ;

skos:hasTopConcept smls-skos:Event ;

skos:hasTopConcept smls-skos:RequirementSourceType ;

skos:hasTopConcept smls-skos:RequirementSeverityType ;

skos:hasTopConcept smls-skos:begins ;

skos:hasTopConcept smls-skos:ends ;

skos:hasTopConcept smls-skos:triggers ;

skos:hasTopConcept smls-skos:transforms ;

skos:hasTopConcept smls-skos:performs ;

skos:hasTopConcept smls-skos:describes ;

skos:hasTopConcept smls-skos:hasState ;

skos:hasTopConcept smls-skos:isImplementedBy ;

skos:hasTopConcept smls-skos:isRealizedBy ;

skos:hasTopConcept smls-skos:hasRequirement ;

skos:hasTopConcept smls-skos:hasRequirementSpecification ;

skos:hasTopConcept smls-skos:hasRequirementSource ;

skos:hasTopConcept smls-skos:hasRequirementSeverity ;

.

smls-skos:Requirement

a skos:Concept ;

skos:broader smls-skos:InformationObject ;

skos:related smls-skos:hasRequirementSpecification ;

skos:related smls-skos:hasRequirementSource ;

skos:related smls-skos:hasRequirementSeverity ;

skos:definition "A rule that should hold for something, defined w.r.t. to its attributes or relations" ;

skos:example "The weight of a bridge should no exceed 100 tons"@en ;

skos:prefLabel "Requirement"@en ;

.

smls-skos:RequirementSeverityType

a skos:Concept ;

skos:prefLabel "Requirement severity type"@en ;

.

smls-skos:RequirementSourceType

a skos:Concept ;

skos:prefLabel "Requirement source type"@en ;

.

smls-skos:hasRequirement

a skos:Concept ;

skos:prefLabel "has requirement"@en ;

.

smls-skos:hasRequirementSpecification

a skos:Concept ;

skos:prefLabel "has requirement specification"@en ;

.

smls-skos:hasRequirementSeverity

a skos:Concept ;

skos:prefLabel "has requirement severity"@en ;

.

smls-skos:hasRequirementSource

a skos:Concept ;

skos:prefLabel "has requirement source"@en ;

.

smls-skos:Activity

a skos:Concept ;

skos:broader smls-skos:Entity ;

skos:definition "An activity is something possibly or actual happens in space and time"@en ;

skos:example "Fluid pressure measurement, driving a car are examples of an <activity>"@en ;

skos:prefLabel "Activity"@en ;

skos:scopeNote "Process, function, human activity, machine activity, task, procedure, project are subclasses of <Activity>"@en ;

.

smls-skos:Function

a skos:Concept ;

skos:broader smls-skos:Activity ;

skos:definition "An activity performed by a functional object"@en ;

skos:example "Pumping, connecting and producing are examples of a <Function>"@en ;

skos:prefLabel "Function"@en ;

.

smls-skos:Event

a skos:Concept ;

skos:definition "A thing that happens or takes place and marks the beginning or ending of a state"@en ;

skos:example "The connection of power to pump P\_101, the take-off of a aeroplane are examples of an <Event>"@en ;

skos:prefLabel "Event"@en ;

skos:scopeNote "Transition, cause, effect, risk, milestone, issue, accident, point in time are subclasses of <Event>"@en ;

.

smls-skos:Object

a skos:Concept ;

skos:broader smls-skos:Entity ;

skos:prefLabel "Object"@en ;

skos:scopeNote "<PhysicalObject> and <InformationObject> are subclasses of <Object>"@en ;

.

smls-skos:Entity

a skos:Concept ;

skos:prefLabel "Entity"@en ;

skos:scopeNote "<Object> and <Activity> are subclasses of <Entity>"@en ;

.

smls-skos:InformationObject

a skos:Concept ;

skos:broader smls-skos:Object ;

skos:definition "Thing that is a whole of information on itself and has an own identity"@en ;

skos:example "A file within a computer system, the PDF file with URI http://material-certificate/M-101-1234.pdf are examples of <InformationObject>"@en ;

skos:prefLabel "Information object"@en ;

skos:scopeNote "Requirement, document, statement, E-mail, code, identifier are subclasses of <InformationObject>"@en ;

.

smls-skos:PhysicalObject

a skos:Concept ;

skos:broader smls-skos:Object ;

skos:definition "Is something that possibly or actually exists in space and time, perceivable through the senses"@en ;

skos:example "Pump P\_101, a single living organism are examples of a <PhysicalObject>"@en ;

skos:prefLabel "Physical object"@en ;

skos:scopeNote "Artefact, matter, person, organisation, stream, connection are subclasses of <PhysicalObject>"@en ;

.

smls-skos:State

a skos:Concept ;

skos:definition "A particular condition that something is in during a specific period in time"@en ;

skos:example "Well-functioning of a car, the condition of an elevator of being un-safe are examples of a <State>"@en ;

skos:prefLabel "State"@en ;

skos:scopeNote "Condition, failure state, objective are subclasses of <State>"@en ;

.

smls-skos:begins

a skos:Concept ;

skos:related smls-skos:Event ;

skos:related smls-skos:Activity ;

skos:prefLabel "begins"@en ;

skos:definition "This relationship applies between an Event and a State."@en ;

skos:example "The lightning strike (event) is the start of the forest fire (state)"@en ;

.

smls-skos:PlannedObject

a skos:Concept ;

skos:broader smls-skos:PhysicalObject ;

skos:prefLabel "Planned object"@en ;

skos:definition "Is something that possibly (not actually) exists in space and time"@en ;

.

smls-skos:BuiltObject

a skos:Concept ;

skos:broader smls-skos:PhysicalObject ;

skos:prefLabel "Built object"@en ;

skos:definition "Is something that actually exists in space and time"@en ;

.

smls-skos:FunctionalObject

a skos:Concept ;

skos:broader smls-skos:PhysicalObject ;

skos:prefLabel "Functional object"@en ;

skos:definition "Is something that possibly or actually exists in space and time, from a functional perspective"@en ;

.

smls-skos:TechnicalObject

a skos:Concept ;

skos:broader smls-skos:PhysicalObject ;

skos:prefLabel "Technical object"@en ;

skos:definition "Is something that possibly or actually exists in space and time, from a technical perspective"@en ;

.

smls-skos:describes

a skos:Concept ;

skos:related smls-skos:InformationObject ;

skos:related smls-skos:PhysicalObject ;

skos:related smls-skos:Activity ;

skos:related smls-skos:Event ;

skos:related smls-skos:State ;

skos:related smls-skos:InformationObject ;

skos:prefLabel "describes"@en ;

skos:definition "This relationship applies between an Information Object and a Physical Object, an Activity, an Event, a State, a Time Period, a Spatial Region or another Information Object."@en ;

skos:example "Take the Ketel bridge. Over time, a dossier has been created of drawings, specifications, reports, invoices, in analogue or in digital form. Each of these documents describes the Ketelbrug and can be seen as an instance of an Information Object that describes the Ketelbrug"@en ;

.

smls-skos:ends

a skos:Concept ;

skos:related smls-skos:Event ;

skos:related smls-skos:Activity ;

skos:prefLabel "ends"@en ;

skos:definition "This relationship applies between an Event and a State."@en ;

skos:example "The thunderstorm (event) ended the nice weather (state)"@en ;

.

smls-skos:hasState

a skos:Concept ;

skos:related smls-skos:PhysicalObject ;

skos:related smls-skos:Activity ;

skos:related smls-skos:State ;

skos:prefLabel "has state"@en ;

skos:definition "This relationship applies between an Activity and a State or between a Physical Object and a State."@en ;

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

.

smls-skos:performs

a skos:Concept ;

skos:related smls-skos:PhysicalObject ;

skos:related smls-skos:Activity ;

skos:prefLabel "performs"@en ;

skos:definition "This relationship applies between a Physical Object and an Activity."@en ;

skos:example "The welding of a bridge railing (an activity) is carried out with the aid of a welding machine (physical object) and the welder (physical object)"@en ;

.

smls-skos:transforms

a skos:Concept ;

skos:related smls-skos:Activity ;

skos:related smls-skos:PhysicalObject ;

skos:prefLabel "transforms"@en ;

skos:definition "This relationship applies between an Activity and a Physical Object or between an Activity and an Information Object."@en ;

skos:example "Baking (an activity) transforms a pizza (a physical object) from unbaked (a state) to baked (another state)"@en ;

.

smls-skos:triggers

a skos:Concept ;

skos:related smls-skos:PhysicalObject ;

skos:related smls-skos:Activity ;

skos:related smls-skos:Event ;

skos:prefLabel "triggers"@en ;

skos:definition "This relationship applies between an Event and an Activity and an Event and a Physical Object."@en ;

skos:example "Take a traffic accident (an event). This initiates the arrival of the emergency services (an activity)"@en ;

.

smls-skos:isImplementedBy

a skos:Concept ;

skos:prefLabel "is implemented by"@en ;

skos:related smls-skos:FunctionalObject ;

skos:related smls-skos:TechnicalObject ;

skos:definition "This relationship applies between a FunctionalObject and a TechnicalObject. The technical object fullfils or play the role of the functional object"@en ;

.

smls-skos:isRealizedBy

a skos:Concept ;

skos:prefLabel "is realized by"@en ;

skos:related smls-skos:PlannedObject ;

skos:related smls-skos:BuiltObject ;

skos:definition "This relationship applies between a PlannedObject and a BuiltObject. The built object corresponds to the planned object"@en ;

.

smls-skos:Demand

a skos:Concept ;

skos:prefLabel "Demand"@en ;

skos:broader smls-skos:RequirementSeverityType ;

.

smls-skos:Wish

a skos:Concept ;

skos:prefLabel "Wish"@en ;

skos:broader smls-skos:RequirementSeverityType ;

.

smls-skos:ByDefinition

a skos:Concept ;

skos:prefLabel "ByDefinition"@en ;

skos:broader smls-skos:RequirementSourceType ;

.

smls-skos:ByClient

a skos:Concept ;

skos:prefLabel "ByClient"@en ;

skos:broader smls-skos:RequirementSourceType ;

.

smls-skos:ByLawOrRegulation

a skos:Concept ;

skos:prefLabel "ByLawOrRegulation"@en ;

skos:broader smls-skos:RequirementSourceType ;

.

smls-skos:ByRecommendation

a skos:Concept ;

skos:prefLabel "ByRecommendation"@en ;

skos:broader smls-skos:RequirementSourceType ;

.

* 1. in RDFS (Turtle format)

# baseURI: https://w3id.org/def/smls-rdfs

# imports: http://qudt.org/schema/qudt

# imports: http://qudt.org/vocab/quantitykind

# imports: http://qudt.org/vocab/unit

# imports: http://www.w3.org/2004/02/skos/core

# prefix: smls-rdfs

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .

@prefix qudt: <http://qudt.org/schema/qudt/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix smls-rdfs: <https://w3id.org/def/smls-rdfs#> .

@prefix unit: <http://qudt.org/vocab/unit/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/def/smls-rdfs>

rdf:type owl:Ontology ;

owl:imports <http://qudt.org/schema/qudt> ;

owl:imports <http://qudt.org/vocab/quantitykind> ;

owl:imports <http://qudt.org/vocab/unit> ;

owl:imports <http://www.w3.org/2004/02/skos/core> ;

.

smls-rdfs:Activity

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:Entity ;

skos:definition "An activity is something possibly or actual happens in space and time"@en ;

skos:example "Fluid pressure measurement, driving a car are examples of an <activity>"@en ;

skos:prefLabel "Activity"@en ;

skos:scopeNote "Process, function, human activity, machine activity, task, procedure, project are subclasses of <Activity>."@en ;

.

smls-rdfs:AddressSystem

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:SpatialReferenceSystem ;

skos:prefLabel "Address system"@en ;

.

smls-rdfs:CalendarSystem

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:TemporalReferenceSystem ;

skos:prefLabel "Calendar system"@en ;

.

smls-rdfs:CoordinateReferenceSystem

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:SpatialReferenceSystem ;

skos:prefLabel "Coordinate reference system"@en ;

.

smls-rdfs:Entity

rdf:type rdfs:Class ;

skos:prefLabel "Entity"@en ;

skos:scopeNote "<Object> and <Activity> are subclasses of <Entity>."@en ;

.

smls-rdfs:EnumerationType

rdf:type rdfs:Class ;

skos:definition "The superclass of all user-defined enumeration classes where the allowed enumeration items are individuals"@en ;

skos:example "LoadLevelType being Low, Medium or High"@en ;

skos:prefLabel "Enumeration type"@en ;

.

smls-rdfs:Event

rdf:type rdfs:Class ;

skos:definition "A thing that happens or takes place and marks the beginning or ending a state"@en ;

skos:example "The connection of power to pump P\_101, the take-off of a aeroplane are examples of an <Event>"@en ;

skos:prefLabel "Event"@en ;

skos:scopeNote "Transition, cause, effect, risk, milestone, issue, accident, point in time are subclasses of <Event>"@en ;

.

smls-rdfs:Function

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:Activity ;

skos:definition "An activity performed by a functional object"@en ;

skos:example "Pumping, connecting and producing are examples of a <Function>"@en ;

skos:prefLabel "Function"@en ;

.

smls-rdfs:FunctionalObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:PhysicalObject ;

skos:prefLabel "Functional object"@en ;

skos:definition "Is something that possibly or actually exists in space and time, from a functional perspective"@en ;

.

smls-rdfs:PlannedObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:PhysicalObject ;

skos:prefLabel "Planned object"@en ;

skos:definition "Is something that possibly (not actually) exists in space and time"@en ;

.

smls-rdfs:IndexKnownSpatialLocations

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:SpatialReferenceSystem ;

skos:prefLabel "Index known spatial locations"@en ;

.

smls-rdfs:InformationObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:Object ;

skos:definition "Thing that is a whole of information on itself and has an own identity"@en ;

skos:example "A file within a computer system, the PDF file with URI http://material-certificate/M-101-1234.pdf are examples of <InformationObject>"@en ;

skos:prefLabel "Information object"@en ;

skos:scopeNote "Requirement, document, statement, E-mail, code, identifier are subclasses of <InformationObject>"@en ;

.

smls-rdfs:LinearReferenceSystem

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:SpatialReferenceSystem ;

skos:prefLabel "Linear reference system"@en ;

.

smls-rdfs:Object

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:Entity ;

skos:prefLabel "Object"@en ;

skos:scopeNote "<PhysicalObject> and <InformationObject> are subclasses of <Object>"@en ;

.

smls-rdfs:PhysicalObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:Object ;

skos:definition "Is something that possibly or actually exists in space and time, perceivable through the senses" ;

skos:example "Pump P\_101, a single living organism are examples of a <PhysicalObject>"@en ;

skos:prefLabel "Physical object"@en ;

skos:scopeNote "Artefact, matter, person, organisation, stream, connection are subclasses of <PhysicalObject>"@en ;

.

smls-rdfs:QuantityValue

rdf:type rdfs:Class ;

skos:prefLabel "Quantity value"@en ;

.

smls-rdfs:BuiltObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:PhysicalObject ;

skos:prefLabel "Built object"@en ;

skos:definition "Is something that actually exists in space and time"@en ;

.

smls-rdfs:SpatialLocation

rdf:type rdfs:Class ;

skos:prefLabel "Spatial location"@en ;

.

smls-rdfs:SpatialReference

rdf:type rdfs:Class ;

skos:prefLabel "Spatial reference"@en ;

.

smls-rdfs:SpatialReferenceSystem

rdf:type rdfs:Class ;

skos:prefLabel "Spatial reference system"@en ;

.

smls-rdfs:SpatialRegion

rdf:type rdfs:Class ;

skos:definition "Demarcated space"@en ;

skos:example "An IFC BREP or GeoSPARQL geometry representing a physical space, as special case of a physical object, such as an office space, a construction area for a building, a maritime traffic zone within the Channel, a hazard zone within a plant or the loading gauge of a train are examples of a <SpatialRegion>"@en ;

skos:prefLabel "Spatial region"@en ;

.

smls-rdfs:State

rdf:type rdfs:Class ;

skos:definition "A particular condition that something is in during a specific period in time"@en ;

skos:example "Well-functioning of a car, the condition of an elevator of being un-safe are examples of a <State>"@en ;

skos:prefLabel "State"@en ;

skos:scopeNote "Condition, failure state, objective are subclasses of <State>"@en ;

.

smls-rdfs:TechnicalObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:PhysicalObject ;

skos:prefLabel "Technical object"@en ;

skos:definition "Is something that possibly or actually exists in space and time, from a technical perspective"@en ;

.

smls-rdfs:TemporalCoordinateReferenceSystem

rdf:type rdfs:Class ;

rdfs:subClassOf smls-rdfs:TemporalReferenceSystem ;

skos:prefLabel "Temporal coordinate reference system"@en ;

.

smls-rdfs:TemporalLocation

rdf:type rdfs:Class ;

skos:prefLabel "Temporal location"@en ;

.

smls-rdfs:TemporalReference

rdf:type rdfs:Class ;

skos:prefLabel "Temporal reference"@en ;

.

smls-rdfs:TemporalReferenceSystem

rdf:type rdfs:Class ;

skos:prefLabel "Temporal reference system"@en ;

.

smls-rdfs:TemporalRegion

rdf:type rdfs:Class ;

skos:definition "A length or portion of time"@en ;

skos:example "Week 12 in 2019, design stage of the Schiphol tunnel are examples of a <TemporalRegion>"@en ;

skos:prefLabel "Temporal region"@en ;

skos:scopeNote "Week, day, life cycle stage are subclasses of <TemporalRegion>"@en ;

.

smls-rdfs:Requirement

a rdfs:Class ;

rdfs:subClassOf smls-rdfs:InformationObject ;

skos:definition "A rule that should hold for something, defined w.r.t. to its attributes or relations" ;

skos:example "The weight of a bridge should no exceed 100 tons"@en ;

skos:prefLabel "Requirement"@en ;

.

smls-rdfs:RequirementSeverityType

a rdfs:Class ;

rdfs:subClassOf smls-rdfs:EnumerationType ;

skos:prefLabel "Requirement severity type"@en ;

.

smls-rdfs:Wish

a smls-rdfs:RequirementSeverityType ;

.

smls-rdfs:Demand

a smls-rdfs:RequirementSeverityType ;

.

smls-rdfs:RequirementSourceType

a rdfs:Class ;

rdfs:subClassOf smls-rdfs:EnumerationType ;

skos:prefLabel "Requirement source type"@en ;

.

smls-rdfs:ByClient

a smls-rdfs:RequirementSourceType ;

.

smls-rdfs:ByDefinition

a smls-rdfs:RequirementSourceType ;

.

smls-rdfs:ByLawOrRegulation

a smls-rdfs:RequirementSourceType ;

.

smls-rdfs:ByRecommendation

a smls-rdfs:RequirementSourceType ;

.

smls-rdfs:hasRequirement

a rdf:Property ;

rdfs:range smls-rdfs:Requirement ;

skos:prefLabel "has requirement"@en ;

.

smls-rdfs:hasRequirementSpecification

a owl:DatatypeProperty ;

rdfs:domain smls-rdfs:Requirement ;

rdfs:range xsd:string;

skos:prefLabel "has requirement specification"@en ;

.

smls-rdfs:hasRequirementSeverity

a rdf:Property ;

rdfs:domain smls-rdfs:Requirement ;

rdfs:range smls-rdfs:RequirementSeverityType ;

skos:prefLabel "has requirement severity"@en ;

.

smls-rdfs:hasRequirementSource

a rdf:Property ;

rdfs:domain smls-rdfs:Requirement ;

rdfs:range smls-rdfs:RequirementSourceType ;

skos:prefLabel "has requirement source"@en ;

.

smls-rdfs:abbreviation

rdf:type rdf:Property ;

rdfs:subPropertyOf skos:altLabel ;

skos:prefLabel "abbreviation"@en ;

.

smls-rdfs:begins

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:Event ;

rdfs:range smls-rdfs:State ;

skos:definition "This relationship applies between an Event and a State"@en ;

skos:example "The lightning strike (event) is the start of the forest fire (state)"@en ;

skos:prefLabel "begins"@en ;

.

smls-rdfs:describes

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:InformationObject ;

skos:definition "This relationship applies between an Information Object and a Physical Object, an Activity, an Event, a State, a Time Period, a Spatial Region or another Information Object"@en ;

skos:example "Take the Ketel bridge. Over time, a dossier has been created of drawings, specifications, reports, invoices, in analogue or in digital form. Each of these documents describes the Ketelbrug and can be seen as an instance of an Information Object that describes the Ketelbrug"@en ;

skos:prefLabel "describes"@en ;

.

smls-rdfs:ends

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:Event ;

rdfs:range smls-rdfs:State ;

skos:definition "This relationship applies between an Event and a State"@en ;

skos:example "The thunderstorm (event) ended the nice weather (state)"@en ;

skos:prefLabel "ends"@en ;

.

smls-rdfs:hasBoundary

rdf:type rdf:Property ;

rdfs:range smls-rdfs:SpatialRegion ;

skos:definition "This relationship applies between a Physical Object and a Spatial Region"@en ;

skos:example "A parcel is bounded by a 1D GML polyline"@en ;

skos:prefLabel "has boundary"@en ;

.

smls-rdfs:hasInterior

rdf:type rdf:Property ;

rdfs:range smls-rdfs:SpatialRegion ;

skos:definition "This relationship applies between a Physical Object and a (topological) Spatial Region, between an Activity and a Spatial Region and between a State and a Spatial Region"@en ;

skos:example "An IfcSpace as physical space can have a Constructive Solid Geometry (CSG) representation as a special case of a 3D region in space"@en ;

skos:prefLabel "has interior"@en ;

.

smls-rdfs:hasPart

rdf:type rdf:Property ;

skos:definition "This relationship applies between Physical Objects themselves, between Information Objects themselves or between Activities themselves"@en ;

skos:example "Ellen's bicycle (physical object) consists of a frame, two wheels and various other parts (all physical objects)"@en ;

skos:prefLabel "has part"@en ;

.

smls-rdfs:hasPeriod

rdf:type rdf:Property ;

rdfs:range smls-rdfs:TemporalRegion ;

skos:definition "This relationship applies between a Condition and a Time Period or between a Physical Object and a Time Period"@en ;

skos:example "From Friday 26 July up to and including Sunday 18 August 2019 (time period) the N231 (physical object) is completely closed off (condition) due to major maintenance (activity)"@en ;

skos:prefLabel "has period"@en ;

.

smls-rdfs:hasSpatialLocation

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:SpatialRegion ;

rdfs:range smls-rdfs:SpatialLocation ;

skos:prefLabel "has spatial location"@en ;

.

smls-rdfs:hasSpatialReference

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:SpatialLocation ;

rdfs:range smls-rdfs:SpatialReference ;

skos:prefLabel "has spatial reference"@en ;

.

smls-rdfs:hasSpatialReferenceSystem

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:SpatialLocation ;

rdfs:range smls-rdfs:SpatialReferenceSystem ;

skos:prefLabel "has spatial reference system"@en ;

.

smls-rdfs:hasState

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:Entity ;

rdfs:range smls-rdfs:State ;

skos:definition "This relationship applies between an Activity and a State or between a Physical Object and a State"@en ;

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

skos:prefLabel "has state"@en ;

.

smls-rdfs:hasTemporalLocation

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:TemporalRegion ;

rdfs:range smls-rdfs:TemporalLocation ;

skos:prefLabel "has temporal location"@en ;

.

smls-rdfs:hasTemporalReference

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:TemporalLocation ;

rdfs:range smls-rdfs:TemporalReference ;

skos:prefLabel "has temporal reference"@en ;

.

smls-rdfs:hasTemporalReferenceSystem

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:TemporalLocation ;

rdfs:range smls-rdfs:TemporalReferenceSystem ;

skos:prefLabel "has temporal reference system"@en ;

.

smls-rdfs:isImplementedBy

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:FunctionalObject ;

rdfs:range smls-rdfs:TechnicalObject ;

skos:prefLabel "is implemented by"@en ;

skos:definition "This relationship applies between a FunctionalObject and a TechnicalObject. The technical object fullfils or play the role of the functional object"@en ;

.

smls-rdfs:isRealizedBy

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:PlannedObject ;

rdfs:range smls-rdfs:BuiltObject ;

skos:prefLabel "is realized by"@en ;

skos:definition "This relationship applies between a PlannedObject and a BuiltObject. The built object corresponds to the planned object"@en ;

.

smls-rdfs:performs

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:PhysicalObject ;

rdfs:range smls-rdfs:Activity ;

skos:definition "This relationship applies between a Physical Object and an Activity"@en ;

skos:example "The welding of a bridge railing (an activity) is carried out with the aid of a welding machine (physical object) and the welder (physical object)"@en ;

skos:prefLabel "performs"@en ;

.

smls-rdfs:quantityKind

rdf:type rdf:Property ;

rdfs:range qudt:QuantityKind ;

skos:prefLabel "quantity kind"@en ;

.

smls-rdfs:transforms

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:Activity ;

rdfs:range smls-rdfs:Object ;

skos:definition "This relationship applies between an Activity and a Physical Object or between an Activity and an Information Object"@en ;

skos:example "Baking (an activity) transforms a pizza (a physical object) from unbaked (a state) to baked (another state)"@en ;

skos:prefLabel "transforms"@en ;

.

smls-rdfs:triggers

rdf:type rdf:Property ;

rdfs:domain smls-rdfs:Entity ;

rdfs:range smls-rdfs:Event ;

skos:definition "This relationship applies between an Event and an Activity and an Event and a Physical Object"@en ;

skos:example "Take a traffic accident (an event). This initiates the arrival of the emergency services (an activity)"@en ;

skos:prefLabel "triggers"@en ;

.

smls-rdfs:unit

rdf:type rdf:Property ;

rdfs:range qudt:Unit ;

skos:prefLabel "unit"@en ;

.

* 1. in OWL (Turtle format)

# baseURI: https://w3id.org/def/smls-owl

# imports: http://qudt.org/schema/qudt

# imports: http://qudt.org/vocab/quantitykind

# imports: http://qudt.org/vocab/unit

# imports: http://www.w3.org/2004/02/skos/core

# prefix: smls-owl

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .

@prefix qudt: <http://qudt.org/schema/qudt/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix smls-owl: <https://w3id.org/def/smls-owl#> .

@prefix unit: <http://qudt.org/vocab/unit/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/def/smls-owl>

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> ;

.

smls-owl:Activity

a owl:Class ;

rdfs:subClassOf smls-owl:Entity ;

rdfs:subClassOf [

a owl:Restriction ;

owl:allValuesFrom smls-owl:Activity ;

owl:onProperty smls-owl:hasPart ;

] ;

skos:definition "An activity is something possibly or actual happens in space and time"@en ;

skos:example "Fluid pressure measurement, driving a car are examples of an <activity>"@en ;

skos:prefLabel "Activity"@en ;

skos:scopeNote "Process, function, human activity, machine activity, task, procedure, project are subclasses of <Activity>"@en ;

.

smls-owl:AddressSystem

a owl:Class ;

rdfs:subClassOf smls-owl:SpatialReferenceSystem ;

skos:prefLabel "Address system"@en ;

.

smls-owl:ByClient

a smls-owl:RequirementSourceType ;

.

smls-owl:ByDefinition

a smls-owl:RequirementSourceType ;

.

smls-owl:ByLawOrRegulation

a smls-owl:RequirementSourceType ;

.

smls-owl:ByRecommendation

a smls-owl:RequirementSourceType ;

.

smls-owl:CalendarSystem

a owl:Class ;

rdfs:subClassOf smls-owl:TemporalReferenceSystem ;

skos:prefLabel "Calendar system"@en ;

.

smls-owl:CoordinateReferenceSystem

a owl:Class ;

rdfs:subClassOf smls-owl:SpatialReferenceSystem ;

skos:prefLabel "Coordinate reference system"@en ;

.

smls-owl:Demand

a smls-owl:RequirementSeverityType ;

.

smls-owl:Entity

a owl:Class ;

skos:prefLabel "Entity"@en ;

skos:scopeNote "<Object> and <Activity> are subclasses of <Entity>"@en ;

.

smls-owl:EnumerationType

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:definition "The superclass of all user-defined enumeration classes where the allowed enumeration items are individuals"@en ;

skos:example "LoadLevelType being Low, Medium or High"@en ;

skos:prefLabel "Enumeration type"@en ;

.

smls-owl:Event

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:definition "A thing that happens or takes place and marks the beginning or ending of a state"@en ;

skos:example "The connection of power to pump P\_101, the take-off of a aeroplane are examples of an <Event>"@en ;

skos:prefLabel "Event"@en ;

skos:scopeNote "Transition, cause, effect, risk, milestone, issue, accident, point in time are subclasses of <Event>"@en ;

.

smls-owl:Function

a owl:Class ;

rdfs:subClassOf smls-owl:Activity ;

skos:definition "An activity performed by a functional object"@en ;

skos:example "Pumping, connecting and producing are examples of a <Function>"@en ;

skos:prefLabel "Function"@en ;

.

smls-owl:FunctionalObject

a owl:Class ;

rdfs:subClassOf smls-owl:PhysicalObject ;

skos:definition "Is something that possibly or actually exists in space and time, from a functional perspective"@en ;

skos:prefLabel "Functional object"@en ;

.

smls-owl:PlannedObject

a owl:Class ;

rdfs:subClassOf smls-owl:PhysicalObject ;

skos:definition "Is something that possibly (not actually) exists in space and time"@en ;

skos:prefLabel "Planned object"@en ;

.

smls-owl:IndexKnownSpatialLocations

a owl:Class ;

rdfs:subClassOf smls-owl:SpatialReferenceSystem ;

skos:prefLabel "Index known spatial locations"@en ;

.

smls-owl:InformationObject

a owl:Class ;

rdfs:subClassOf smls-owl:Object ;

rdfs:subClassOf [

a owl:Restriction ;

owl:allValuesFrom smls-owl:InformationObject ;

owl:onProperty smls-owl:hasPart ;

] ;

skos:definition "Thing that is a whole of information on itself and has an own identity"@en ;

skos:example "A file within a computer system, the PDF file with URI http://material-certificate/M-101-1234.pdf are examples of <InformationObject>"@en ;

skos:prefLabel "Information object"@en ;

skos:scopeNote "Requirement, document, statement, E-mail, code, identifier are subclasses of <InformationObject>"@en ;

.

smls-owl:LinearReferenceSystem

a owl:Class ;

rdfs:subClassOf smls-owl:SpatialReferenceSystem ;

skos:prefLabel "Linear reference system"@en ;

.

smls-owl:Object

a owl:Class ;

rdfs:subClassOf smls-owl:Entity ;

skos:prefLabel "Object"@en ;

skos:scopeNote "<PhysicalObject> and <InformationObject> are subclasses of <Object>"@en ;

.

smls-owl:PhysicalObject

a owl:Class ;

rdfs:subClassOf smls-owl:Object ;

rdfs:subClassOf [

a owl:Restriction ;

owl:allValuesFrom smls-owl:PhysicalObject ;

owl:onProperty smls-owl:hasPart ;

] ;

skos:definition "Is something that possibly or actually exists in space and time, perceivable through the senses"@en ;

skos:example "Pump P\_101, a single living organism are examples of a <PhysicalObject>"@en ;

skos:prefLabel "Physical object"@en ;

skos:scopeNote "Artefact, matter, person, organisation, stream, connection are subclasses of <PhysicalObject>"@en ;

.

smls-owl:QuantityValue

a owl:Class ;

rdfs:subClassOf owl:Thing ;

rdfs:subClassOf [

a owl:Restriction ;

owl:allValuesFrom xsd:decimal ;

owl:onProperty rdf:value ;

] ;

rdfs:subClassOf [

a owl:Restriction ;

owl:cardinality "1"^^xsd:nonNegativeInteger ;

owl:onProperty rdf:value ;

] ;

skos:prefLabel "Quantity value"@en ;

.

smls-owl:BuiltObject

a owl:Class ;

rdfs:subClassOf smls-owl:PhysicalObject ;

skos:definition "Is something that actually exists in space and time"@en ;

skos:prefLabel "Built object"@en ;

.

smls-owl:Requirement

a owl:Class ;

rdfs:subClassOf smls-owl:InformationObject ;

skos:definition "A rule that should hold for something, defined w.r.t. to its attributes or relations" ;

skos:example "The weight of a bridge should no exceed 100 tons"@en ;

skos:prefLabel "Requirement"@en ;

.

smls-owl:RequirementSeverityType

a owl:Class ;

rdfs:subClassOf smls-owl:EnumerationType ;

skos:prefLabel "Requirement severity type"@en ;

owl:oneOf (

smls-owl:Demand

smls-owl:Wish ) ;

.

smls-owl:RequirementSourceType

a owl:Class ;

rdfs:subClassOf smls-owl:EnumerationType ;

skos:prefLabel "Requirement source type"@en ;

owl:oneOf (

smls-owl:ByDefinition

smls-owl:ByLawOrRegulation

smls-owl:ByRecommendation ) ;

.

smls-owl:SpatialLocation

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:prefLabel "Spatial location"@en ;

.

smls-owl:SpatialReference

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:prefLabel "Spatial reference"@en ;

.

smls-owl:SpatialReferenceSystem

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:prefLabel "Spatial reference system"@en ;

.

smls-owl:SpatialRegion

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:definition "Demarcated space"@en ;

skos:example "An IFC BREP or GeoSPARQL geometry representing a physical space, as special case of a physical object, such as an office space, a construction area for a building, a maritime traffic zone within the Channel, a hazard zone within a plant or the loading gauge of a train are examples of a <SpatialRegion>"@en ;

skos:prefLabel "Spatial region"@en ;

.

smls-owl:State

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:definition "A particular condition that something is in during a specific period in time"@en ;

skos:example "Well-functioning of a car, the condition of an elevator of being un-safe are examples of a <State>"@en ;

skos:prefLabel "State"@en ;

skos:scopeNote "Condition, failure state, objective are subclasses of <State>"@en ;

.

smls-owl:TechnicalObject

a owl:Class ;

rdfs:subClassOf smls-owl:PhysicalObject ;

skos:definition "Is something that possibly or actually exists in space and time, from a technical perspective"@en ;

skos:prefLabel "Technical object"@en ;

.

smls-owl:TemporalCoordinateReferenceSystem

a owl:Class ;

rdfs:subClassOf smls-owl:TemporalReferenceSystem ;

skos:prefLabel "Temporal coordinate reference system"@en ;

.

smls-owl:TemporalLocation

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:prefLabel "Temporal location"@en ;

.

smls-owl:TemporalReference

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:prefLabel "Temporal reference"@en ;

.

smls-owl:TemporalReferenceSystem

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:prefLabel "Temporal reference system"@en ;

.

smls-owl:TemporalRegion

a owl:Class ;

rdfs:subClassOf owl:Thing ;

skos:definition "A length or portion of time" ;

skos:example "Week 12 in 2019, design stage of the Schiphol tunnel are examples of a <TemporalRegion>"@en ;

skos:prefLabel "Temporal region"@en ;

skos:scopeNote "Week, day, life cycle stage are subclasses of <TemporalRegion>"@en ;

.

smls-owl:Wish

a smls-owl:RequirementSeverityType ;

.

smls-owl:abbreviation

a owl:AnnotationProperty ;

rdfs:subPropertyOf skos:altLabel ;

skos:prefLabel "abbreviation"@en ;

.

smls-owl:begins

a owl:ObjectProperty ;

rdfs:domain smls-owl:Event ;

rdfs:range smls-owl:State ;

skos:definition "This relationship applies between an Event and a State"@en ;

skos:example "The lightning strike (event) is the start of the forest fire (state)"@en ;

skos:prefLabel "begins"@en ;

.

smls-owl:describes

a owl:ObjectProperty ;

rdfs:domain smls-owl:InformationObject ;

rdfs:range [

a owl:Class ;

owl:unionOf (

smls-owl:PhysicalObject

smls-owl:Activity

smls-owl:Event

smls-owl:State

smls-owl:InformationObject

) ;

] ;

skos:definition "This relationship applies between an Information Object and a Physical Object, an Activity, an Event, a State, a Time Period, a Spatial Region or another Information Object"@en ;

skos:example "Take the Ketel bridge. Over time, a dossier has been created of drawings, specifications, reports, invoices, in analogue or in digital form. Each of these documents describes the Ketelbrug and can be seen as an instance of an Information Object that describes the Ketelbrug"@en ;

skos:prefLabel "describes"@en ;

.

smls-owl:ends

a owl:ObjectProperty ;

rdfs:domain smls-owl:Event ;

rdfs:range smls-owl:State ;

skos:definition "This relationship applies between an Event and a State"@en ;

skos:example "The thunderstorm (event) ended the nice weather (state)"@en ;

skos:prefLabel "ends"@en ;

.

smls-owl:hasBoundary

a owl:ObjectProperty ;

rdfs:domain [

a owl:Class ;

owl:unionOf (

smls-owl:PhysicalObject

smls-owl:Activity

smls-owl:State

) ;

] ;

rdfs:range smls-owl:SpatialRegion ;

skos:definition "This relationship applies between a Physical Object and a Spatial Region"@en ;

skos:example "A parcel is bounded by a 1D GML polyline"@en ;

skos:prefLabel "has boundary"@en ;

.

smls-owl:hasInterior

a owl:ObjectProperty ;

rdfs:domain [

a owl:Class ;

owl:unionOf (

smls-owl:PhysicalObject

smls-owl:Activity

smls-owl:State

) ;

] ;

rdfs:range smls-owl:SpatialRegion ;

skos:definition "This relationship applies between a Physical Object and a (topological) Spatial Region, between an Activity and a Spatial Region and between a State and a Spatial Region"@en ;

skos:example "An IfcSpace as physical space can have a Constructive Solid Geometry (CSG) representation as a special case of a 3D region in space"@en ;

skos:prefLabel "has interior"@en ;

.

smls-owl:hasPart

a owl:ObjectProperty ;

skos:definition "This relationship applies between Physical Objects themselves, between Information Objects themselves or between Activities themselves"@en ;

skos:example "Ellen's bicycle (physical object) consists of a frame, two wheels and various other parts (all physical objects)"@en ;

skos:prefLabel "has part"@en ;

.

smls-owl:hasPeriod

a owl:ObjectProperty ;

rdfs:domain [

a owl:Class ;

owl:unionOf (

smls-owl:PhysicalObject

smls-owl:Activity

smls-owl:State

) ;

] ;

rdfs:range smls-owl:TemporalRegion ;

skos:definition "This relationship applies between a Condition and a Time Period or between a Physical Object and a Time Period"@en ;

skos:example "From Friday 26 July up to and including Sunday 18 August 2019 (time period) the N231 (physical object) is completely closed off (condition) due to major maintenance (activity)"@en ;

skos:prefLabel "has period"@en ;

.

smls-owl:hasRequirement

a owl:ObjectProperty ;

rdfs:range smls-owl:Requirement ;

skos:prefLabel "has requirement"@en ;

.

smls-owl:hasRequirementSpecification

a owl:DatatypeProperty ;

rdfs:domain smls-owl:Requirement ;

rdfs:range xsd:string;

skos:prefLabel "has requirement specification"@en ;

.

smls-owl:hasRequirementSeverity

a owl:ObjectProperty ;

rdfs:domain smls-owl:Requirement ;

rdfs:range smls-owl:RequirementSeverityType ;

skos:prefLabel "has requirement severity"@en ;

.

smls-owl:hasRequirementSource

a owl:ObjectProperty ;

rdfs:domain smls-owl:Requirement ;

rdfs:range smls-owl:RequirementSourceType ;

skos:prefLabel "has requirement source"@en ;

.

smls-owl:hasSpatialLocation

a owl:ObjectProperty ;

rdfs:domain smls-owl:SpatialRegion ;

rdfs:range smls-owl:SpatialLocation ;

skos:prefLabel "has spatial location"@en ;

.

smls-owl:hasSpatialReference

a owl:ObjectProperty ;

rdfs:domain smls-owl:SpatialLocation ;

rdfs:range smls-owl:SpatialReference ;

skos:prefLabel "has spatial reference"@en ;

.

smls-owl:hasSpatialReferenceSystem

a owl:ObjectProperty ;

rdfs:domain smls-owl:SpatialLocation ;

rdfs:range smls-owl:SpatialReferenceSystem ;

skos:prefLabel "has spatial reference system"@en ;

.

smls-owl:hasState

a owl:ObjectProperty ;

rdfs:domain smls-owl:Entity ;

rdfs:range smls-owl:State ;

skos:definition "This relationship applies between an Activity and a State or between a Physical Object and a State"@en ;

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

skos:prefLabel "has state"@en ;

.

smls-owl:hasTemporalLocation

a owl:ObjectProperty ;

rdfs:domain smls-owl:TemporalRegion ;

rdfs:range smls-owl:TemporalLocation ;

skos:prefLabel "has temporal location"@en ;

.

smls-owl:hasTemporalReference

a owl:ObjectProperty ;

rdfs:domain smls-owl:TemporalLocation ;

rdfs:range smls-owl:TemporalReference ;

skos:prefLabel "has temporal reference"@en ;

.

smls-owl:hasTemporalReferenceSystem

a owl:ObjectProperty ;

rdfs:domain smls-owl:TemporalLocation ;

rdfs:range smls-owl:TemporalReferenceSystem ;

skos:prefLabel "has temporal reference system"@en ;

.

smls-owl:isImplementedBy

a owl:ObjectProperty ;

rdfs:domain smls-owl:FunctionalObject ;

rdfs:range smls-owl:TechnicalObject ;

skos:definition "This relationship applies between a FunctionalObject and a TechnicalObject. The technical object fullfils or play the role of the functional object"@en ;

skos:prefLabel "is implemented by"@en ;

.

smls-owl:isRealizedBy

a owl:ObjectProperty ;

rdfs:domain smls-owl:PlannedObject ;

rdfs:range smls-owl:BuiltObject ;

skos:definition "This relationship applies between a PlannedObject and a BuiltObject. The real object corresponds to the planned object"@en ;

skos:prefLabel "is realized by"@en ;

.

smls-owl:performs

a owl:ObjectProperty ;

rdfs:domain smls-owl:PhysicalObject ;

rdfs:range smls-owl:Activity ;

skos:definition "This relationship applies between a Physical Object and an Activity"@en ;

skos:example "The welding of a bridge railing (an activity) is carried out with the aid of a welding machine (physical object) and the welder (physical object)"@en ;

skos:prefLabel "performs"@en ;

.

smls-owl:quantityKind

a owl:ObjectProperty ;

rdfs:range qudt:QuantityKind ;

skos:prefLabel "quantity kind"@en ;

.

smls-owl:transforms

a owl:ObjectProperty ;

rdfs:domain smls-owl:Activity ;

rdfs:range smls-owl:Object ;

skos:definition "This relationship applies between an Activity and a Physical Object or between an Activity and an Information Object"@en ;

skos:example "Baking (an activity) transforms a pizza (a physical object) from unbaked (a state) to baked (another state)"@en ;

skos:prefLabel "transforms"@en ;

.

smls-owl:triggers

a owl:ObjectProperty ;

rdfs:domain smls-owl:Entity ;

rdfs:range smls-owl:Event ;

skos:definition "This relationship applies between an Event and an Activity and an Event and a Physical Object"@en ;

skos:example "Take a traffic accident (an event). This initiates the arrival of the emergency services (an activity)"@en ;

skos:prefLabel "triggers"@en ;

.

smls-owl:unit

a owl:ObjectProperty ;

rdfs:range qudt:Unit ;

skos:prefLabel "unit"@en ;

.

[

a owl:AllDisjointClasses ;

owl:members (

smls-owl:FunctionalObject

smls-owl:TechnicalObject

) ;

].

[

a owl:AllDisjointClasses ;

owl:members (

smls-owl:ImaginaryObject

smls-owl:RealObject

) ;

].

[

a owl:AllDisjointClasses ;

owl:members (

smls-owl:Entity

smls-owl:State

smls-owl:Event

smls-owl:TemporalRegion

smls-owl:SpatialRegion

) ;

].

[

a owl:AllDisjointClasses ;

owl:members (

smls-owl:Object

smls-owl:Activity

) ;

].

[

a owl:AllDisjointClasses ;

owl:members (

smls-owl:PhysicalObject

smls-owl:InformationObject

) ;

].

* 1. in SHACL (Turtle format)

# baseURI: https://w3id.org/def/smls-shacl

# imports: http://qudt.org/schema/qudt

# imports: http://qudt.org/vocab/quantitykind

# imports: http://qudt.org/vocab/unit

# imports: http://www.w3.org/2004/02/skos/core

# imports: http://datashapes.org/dash

# prefix: smls-shacl

@prefix smls-shacl: <https://w3id.org/def/smls-shacl#> .

@prefix dash: <http://datashapes.org/dash#> .

@prefix sh: <http://www.w3.org/ns/shacl#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .

@prefix qudt: <http://qudt.org/schema/qudt/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix unit: <http://qudt.org/vocab/unit/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/def/smls-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://datashapes.org/dash> ;

.

smls-shacl:Requirement

a rdfs:Class ;

rdfs:subClassOf smls-shacl:InformationObject ;

skos:definition "A rule that should hold for something, defined w.r.t. to its attributes or relations" ;

skos:example "The weight of a bridge should no exceed 100 tons"@en ;

skos:prefLabel "Requirement"@en ;

.

smls-shacl:RequirementShape

a sh:NodeShape ;

sh:targetClass smls-shacl:Requirement ;

sh:property [

sh:path smls-shacl:hasRequirementSpecification ;

sh:datatype xsd:string ;

] ;

sh:property [

sh:path smls-shacl:hasRequirementSource ;

sh:class smls-shacl:RequirementsSourceType ;

sh:in (smls-shacl:Demand smls-shacl:Wish ) ;

] ;

sh:property [

sh:path smls-shacl:hasRequirementSeverity ;

sh:class smls-shacl:RequirementsSeverityType ;

sh:in (smls-shacl:ByDefinition smls-shacl:ByClient smls-shacl:ByLawOrRegulation smls-shacl:ByRecommendation ) ;

] ;

.

smls-shacl:RequirementSeverityType

a rdfs:Class ;

rdfs:subClassOf smls-shacl:EnumerationType ;

skos:prefLabel "Requirement severity type"@en ;

.

smls-shacl:Wish

a smls-shacl:RequirementSeverityType ;

.

smls-shacl:Demand

a smls-shacl:RequirementSeverityType ;

.

smls-shacl:RequirementSourceType

a rdfs:Class ;

rdfs:subClassOf smls-shacl:EnumerationType ;

skos:prefLabel "Requirement source type"@en ;

.

smls-shacl:ByClient

a smls-shacl:RequirementSourceType ;

.

smls-shacl:ByDefinition

a smls-shacl:RequirementSourceType ;

.

smls-shacl:ByLawOrRegulation

a smls-shacl:RequirementSourceType ;

.

smls-shacl:ByRecommendation

a smls-shacl:RequirementSourceType ;

.

smls-shacl:hasRequirement

a rdf:Property ;

skos:prefLabel "has requirement"@en ;

.

smls-shacl:hasRequirementSpecification

a rdf:Property ;

skos:prefLabel "has requirement specification"@en ;

.

smls-shacl:hasRequirementShape

a sh:NodeShape ;

sh:targetObjectsOf smls-shacl:hasRequirement ;

sh:class smls-shacl:Requirement ;

.

smls-shacl:hasRequirementSeverity

a rdf:Property ;

skos:prefLabel "has requirement severity"@en ;

.

smls-shacl:hasRequirementSource

a rdf:Property ;

skos:prefLabel "has requirement source"@en ;

.

smls-shacl:Activity

a rdfs:Class ;

rdfs:subClassOf smls-shacl:Entity ;

skos:definition "An activity is something possibly or actually happens in space and time"@en ;

skos:example "Fluid pressure measurement, driving a car are examples of an <activity>"@en ;

skos:prefLabel "Activity"@en ;

skos:scopeNote "Process, function, human activity, machine activity, task, procedure, project are subclasses of <Activity>"@en ;

.

smls-shacl:Object

a rdfs:Class ;

rdfs:subClassOf smls-shacl:Entity ;

skos:prefLabel "Object"@en ;

skos:scopeNote "<PhysicalObject> and <InformationObject> are subclasses of <Object>"@en ;

.

smls-shacl:ObjectShape

a sh:NodeShape ;

sh:targetClass smls-shacl:Object ;

sh:property [

sh:path smls-shacl:hasPart ;

sh:class smls-shacl:Object ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:Object ;

sh:not [ sh:hasValue smls-shacl:Activity ];

] ;

.

smls-shacl:Entity

a rdfs:Class ;

skos:prefLabel "Entity"@en ;

skos:scopeNote "<Object> and <Activity> are subclasses of <Entity>"@en ;

.

smls-shacl:EntityShape

a sh:NodeShape ;

sh:targetClass smls-shacl:Entity ;

sh:property [

sh:path smls-shacl:triggers ;

sh:class smls-shacl:Event ;

] ;

sh:property [

sh:path smls-shacl:hasState ;

sh:class smls-shacl:State ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:Entity ;

sh:not [ sh:or ( [ sh:hasValue smls-shacl:State ; ]

[ sh:hasValue smls-shacl:Event ; ]

[ sh:hasValue smls-shacl:SpatialRegion ; ]

[ sh:hasValue smls-shacl:TemporalRegion ; ]

) ;

] ; ] ;

.

smls-shacl:Function

a rdfs:Class ;

rdfs:subClassOf smls-shacl:Activity ;

skos:definition "An activity performed by a functional object"@en ;

skos:example "Pumping, connecting and producing are examples of a <Function>"@en ;

skos:prefLabel "Function"@en ;

.

smls-shacl:FunctionalObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-shacl:PhysicalObject ;

skos:prefLabel "Functional object"@en ;

skos:definition "Is something that possibly or actually exists in space and time, from a functional perspective"@en ;

.

smls-shacl:ActivityShape

a sh:NodeShape ;

sh:targetClass smls-shacl:Activity ;

sh:property [

sh:path smls-shacl:hasPart ;

sh:class smls-shacl:Activity ;

] ;

sh:property [

sh:path smls-shacl:hasBoundary ;

sh:class smls-shacl:SpatialRegion ;

] ;

sh:property [

sh:path smls-shacl:hasInterior ;

sh:class smls-shacl:SpatialRegion ;

] ;

sh:property [

sh:path smls-shacl:hasPeriod ;

sh:class smls-shacl:TemporalRegion ;

] ;

sh:property [

sh:path smls-shacl:transforms ;

sh:class smls-shacl:Object ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:Activity ;

sh:not [ sh:or ( [ sh:hasValue smls-shacl:Object ; ]

[ sh:hasValue smls-shacl:State ; ]

[ sh:hasValue smls-shacl:Event ; ]

[ sh:hasValue smls-shacl:SpatialRegion ; ]

[ sh:hasValue smls-shacl:TemporalRegion ; ]

) ;

] ; ] ;

.

smls-shacl:AddressSystem

a rdfs:Class ;

rdfs:subClassOf smls-shacl:SpatialReferenceSystem ;

skos:prefLabel "Address system"@en ;

.

smls-shacl:CalendarSystem

a rdfs:Class ;

rdfs:subClassOf smls-shacl:TemporalReferenceSystem ;

skos:prefLabel "Calendar system"@en ;

.

smls-shacl:CoordinateReferenceSystem

a rdfs:Class ;

rdfs:subClassOf smls-shacl:SpatialReferenceSystem ;

skos:prefLabel "Coordinate reference system"@en ;

.

smls-shacl:EnumerationType

a rdfs:Class ;

skos:definition "The superclass of all user-defined enumeration classes where the allowed enumeration items are individuals"@en ;

skos:example "LoadLevelType being Low, Medium or High"@en ;

skos:prefLabel "Enumeration type"@en ;

.

smls-shacl:Event

a rdfs:Class ;

skos:definition "A thing that happens or takes place and marks the beginning or ending of a state"@en ;

skos:example "The connection of power to pump P\_101, the take-off of a aeroplane are examples of an <Event>"@en ;

skos:prefLabel "Event"@en ;

skos:scopeNote "Transition, cause, effect, risk, milestone, issue, accident, point in time are subclasses of <Event>"@en ;

.

smls-shacl:EventShape

a sh:NodeShape ;

sh:targetClass smls-shacl:Event ;

sh:property [

sh:path smls-shacl:begins ;

sh:class smls-shacl:State ;

] ;

sh:property [

sh:path smls-shacl:ends ;

sh:class smls-shacl:State ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:Activity ;

sh:not [ sh:or ( [ sh:hasValue smls-shacl:Entity ; ]

[ sh:hasValue smls-shacl:State ; ]

[ sh:hasValue smls-shacl:SpatialRegion ; ]

[ sh:hasValue smls-shacl:TemporalRegion ; ]

) ;

] ; ] ;

.

smls-shacl:IndexKnownSpatialLocations

a rdfs:Class ;

rdfs:subClassOf smls-shacl:SpatialReferenceSystem ;

skos:prefLabel "Index known spatial locations"@en ;

.

smls-shacl:InformationObject

a rdfs:Class ;

rdfs:subClassOf smls-shacl:Object ;

skos:definition "Thing that is a whole of information on itself and has an own identity"@en ;

skos:example "A file within a computer system, the PDF file with URI http://material-certificate/M-101-1234.pdf are examples of <InformationObject>"@en ;

skos:prefLabel "Information object"@en ;

skos:scopeNote "Requirement, document, statement, E-mail, code, identifier are subclasses of <InformationObject>"@en ;

.

smls-shacl:PlannedObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-shacl:PhysicalObject ;

skos:prefLabel "Planned object"@en ;

skos:definition "Is something that possibly (not actually) exists in space and time"@en ;

.

smls-shacl:BuiltObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-shacl:PhysicalObject ;

skos:prefLabel "Built object"@en ;

skos:definition "Is something that actually exists in space and time"@en ;

.

smls-shacl:InformationObjectShape

a sh:NodeShape ;

sh:targetClass smls-shacl:InformationObject ;

sh:property [

sh:path smls-shacl:hasPart ;

sh:class smls-shacl:InformationObject ;

] ;

sh:property [

sh:path smls-shacl:describes ;

sh:or ( [ sh:class smls-shacl:PhysicalObject ; ]

[ sh:class smls-shacl:Activity ; ]

[ sh:class smls-shacl:Event ; ]

[ sh:class smls-shacl:State ; ]

[ sh:class smls-shacl:InformationObject ; ]

) ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:InformationObject ;

sh:not [ sh:hasValue smls-shacl:PhysicalObject ] ;

] ;

.

smls-shacl:LinearReferenceSystem

a rdfs:Class ;

rdfs:subClassOf smls-shacl:SpatialReferenceSystem ;

skos:prefLabel "Linear reference system"@en ;

.

smls-shacl:PhysicalObject

a rdfs:Class ;

skos:definition "Is something that possibly or actually exists in space and time, perceivable through the senses"@en ;

skos:example "Pump P\_101, a single living organism are examples of a <PhysicalObject>"@en ;

skos:prefLabel "Physical object"@en ;

skos:scopeNote "Artefact, matter, person, organisation, stream, connection are subclasses of <PhysicalObject>." ;

.

smls-shacl:PhysicalObjectShape

a sh:NodeShape ;

sh:targetClass smls-shacl:PhysicalObject ;

sh:property [

sh:path smls-shacl:hasPart ;

sh:class smls-shacl:PhysicalObject ;

] ;

sh:property [

sh:path smls-shacl:hasBoundary ;

sh:class smls-shacl:SpatialRegion ;

] ;

sh:property [

sh:path smls-shacl:hasInterior ;

sh:class smls-shacl:SpatialRegion ;

] ;

sh:property [

sh:path smls-shacl:hasPeriod ;

sh:class smls-shacl:TemporalRegion ;

] ;

sh:property [

sh:path smls-shacl:performs ;

sh:class smls-shacl:Activity ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:PhysicalObject ;

sh:not [ sh:hasValue smls-shacl:InformationObject ] ;

] ;

.

smls-shacl:QuantityValue

rdf:type rdfs:Class ;

skos:prefLabel "Quantity value"@en ;

.

smls-shacl:QuantityValueShape

a sh:NodeShape ;

sh:targetClass smls-shacl:QuantityValue ;

sh:property [

sh:path rdf:value ;

sh:datatype xsd:decimal ;

sh:minCount 1 ;

sh:maxCount 1 ;

] ;

.

smls-shacl:SpatialLocation

a rdfs:Class ;

skos:prefLabel "Spatial location"@en ;

.

smls-shacl:SpatialLocationShape

a sh:NodeShape ;

sh:targetClass smls-shacl:SpatialLocation ;

sh:property [

sh:path smls-shacl:hasSpatialReference ;

sh:class smls-shacl:SpatialReference ;

] ;

sh:property [

sh:path smls-shacl:hasSpatialReferenceSystem ;

sh:class smls-shacl:SpatialReferenceSystem ;

] ;

.

smls-shacl:SpatialReference

a rdfs:Class ;

skos:prefLabel "Spatial reference"@en ;

.

smls-shacl:SpatialReferenceSystem

a rdfs:Class ;

skos:prefLabel "Spatial reference system"@en ;

.

smls-shacl:SpatialRegion

a rdfs:Class ;

skos:definition "Demarcated space"@en ;

skos:example "An IFC BREP or GeoSPARQL geometry representing a physical space, as special case of a physical object, such as an office space, a construction area for a building, a maritime traffic zone within the Channel, a hazard zone within a plant or the loading gauge of a train are examples of a <SpatialRegion>"@en ;

skos:prefLabel "Spatial region"@en ;

.

smls-shacl:SpatialRegionShape

a sh:NodeShape ;

sh:targetClass smls-shacl:SpatialRegion ;

sh:property [

sh:path smls-shacl:hasSpatialLocation ;

sh:class smls-shacl:SpatialLocation ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:SpatialRegion ;

sh:not [ sh:or ( [ sh:hasValue smls-shacl:Entity ; ]

[ sh:hasValue smls-shacl:State ; ]

[ sh:hasValue smls-shacl:Event ; ]

[ sh:hasValue smls-shacl:TemporalRegion ; ]

) ;

] ; ] ;

.

smls-shacl:State

a rdfs:Class ;

skos:definition "A particular condition that something is in during a specific period in time"@en ;

skos:example "Well-functioning of a car, the condition of an elevator of being un-safe are examples of a <State>"@en ;

skos:prefLabel "State"@en ;

skos:scopeNote "Condition, failure state, objective are subclasses of <State>"@en ;

.

smls-shacl:StateShape

a sh:NodeShape ;

sh:targetClass smls-shacl:State ;

sh:property [

sh:path smls-shacl:hasBoundary ;

sh:class smls-shacl:SpatialRegion ;

] ;

sh:property [

sh:path smls-shacl:hasInterior ;

sh:class smls-shacl:SpatialRegion ;

] ;

sh:property [

sh:path smls-shacl:hasPeriod ;

sh:class smls-shacl:TemporalRegion ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:State ;

sh:not [ sh:or ( [ sh:hasValue smls-shacl:Entity ; ]

[ sh:hasValue smls-shacl:SpatialRegion ; ]

[ sh:hasValue smls-shacl:Event ; ]

[ sh:hasValue smls-shacl:TemporalRegion ; ]

) ;

] ; ] ;

.

smls-shacl:TechnicalObject

rdf:type rdfs:Class ;

rdfs:subClassOf smls-shacl:PhysicalObject ;

skos:prefLabel "Technical object"@en ;

skos:definition "Is something that possibly or actually exists in space and time, from a technical perspective"@en ;

.

smls-shacl:TemporalCoordinateReferenceSystem

a rdfs:Class ;

rdfs:subClassOf smls-shacl:TemporalReferenceSystem ;

skos:prefLabel "Temporal coordinate reference system"@en ;

.

smls-shacl:TemporalLocation

a rdfs:Class ;

skos:prefLabel "Temporal location"@en ;

.

smls-shacl:TemporalLocationShape

a sh:NodeShape ;

sh:targetClass smls-shacl:TemporalLocation ;

sh:property [

sh:path smls-shacl:hasTemporalReference ;

sh:class smls-shacl:TemporalReference ;

] ;

sh:property [

sh:path smls-shacl:hasTemporallReferenceSystem ;

sh:class smls-shacl:TemporalReferenceSystem ;

] ;

.

smls-shacl:TemporalReference

a rdfs:Class ;

skos:prefLabel "Temporal reference"@en ;

.

smls-shacl:TemporalReferenceSystem

a rdfs:Class ;

skos:prefLabel "Temporal reference system"@en ;

.

smls-shacl:TemporalRegion

a rdfs:Class ;

skos:definition "A length or portion of time"@en ;

skos:example "Week 12 in 2019, design stage of the Schiphol tunnel are examples of a <TemporalRegion>"@en ;

skos:prefLabel "Temporal region"@en ;

skos:scopeNote "Week, day, life cycle stage are subclasses of <TemporalRegion>"@en ;

.

smls-shacl:TemporalRegionShape

a sh:NodeShape ;

sh:targetClass smls-shacl:TemporalRegion ;

sh:property [

sh:path smls-shacl:hasTemporalLocation ;

sh:class smls-shacl:TemporalLocation ;

] ;

sh:property [

sh:path ( rdf:type

[ sh:zeroOrMorePath rdfs:subClassOf ; ] ) ;

sh:hasValue smls-shacl:TemporalRegion ;

sh:not [ sh:or ( [ sh:hasValue smls-shacl:Entity ; ]

[ sh:hasValue smls-shacl:State ; ]

[ sh:hasValue smls-shacl:Event ; ]

[ sh:hasValue smls-shacl:SpatialRegion ; ]

) ;

] ; ] ;

.

smls-shacl:abbreviation

a rdf:Property ;

rdfs:subPropertyOf skos:altLabel ;

skos:prefLabel "abbreviation"@en ;

.

smls-shacl:begins

a rdf:Property ;

skos:prefLabel "begins"@en ;

skos:definition "This relationship applies between an Event and a State"@en ;

skos:example "The lightning strike (event) is the start of the forest fire (state)"@en ;

.

smls-shacl:describes

a rdf:Property ;

skos:prefLabel "describes"@en ;

skos:definition "This relationship applies between an Information Object and a Physical Object, an Activity, an Event, a State, a Time Period, a Spatial Region or another Information Object"@en ;

skos:example "Take the Ketel bridge. Over time, a dossier has been created of drawings, specifications, reports, invoices, in analogue or in digital form. Each of these documents describes the Ketelbrug and can be seen as an instance of an Information Object that describes the Ketelbrug"@en ;

.

smls-shacl:ends

a rdf:Property ;

skos:prefLabel "ends"@en ;

skos:definition "This relationship applies between an Event and a State"@en ;

skos:example "The thunderstorm (event) ended the nice weather (state)"@en ;

.

smls-shacl:hasBoundary

a rdf:Property ;

skos:prefLabel "has boundary"@en ;

skos:definition "This relationship applies between a Physical Object and a Spatial Region"@en ;

skos:example "A parcel is bounded by a 1D GML polyline"@en ;

.

smls-shacl:hasInterior

a rdf:Property ;

skos:prefLabel "has interior"@en ;

skos:definition "This relationship applies between a Physical Object and a (topological) Spatial Region, between an Activity and a Spatial Region and between a State and a Spatial Region"@en ;

skos:example "An IfcSpace as physical space can have a Constructive Solid Geometry (CSG) representation as a special case of a 3D region in space"@en ;

.

smls-shacl:hasPart

a rdf:Property ;

skos:prefLabel "has part"@en ;

skos:definition "This relationship applies between Physical Objects themselves, between Information Objects themselves or between Activities themselves"@en ;

skos:example "Ellen's bicycle (physical object) consists of a frame, two wheels and various other parts (all physical objects)"@en ;

.

smls-shacl:hasPeriod

a rdf:Property ;

skos:prefLabel "has period"@en ;

skos:definition "This relationship applies between a Condition and a Time Period or between a Physical Object and a Time Period"@en ;

skos:example "From Friday 26 July up to and including Sunday 18 August 2019 (time period) the N231 (physical object) is completely closed off (condition) due to major maintenance (activity)"@en ;

.

smls-shacl:hasSpatialLocation

a rdf:Property ;

skos:prefLabel "has spatial location"@en ;

.

smls-shacl:hasSpatialReference

a rdf:Property ;

skos:prefLabel "has spatial reference"@en ;

.

smls-shacl:hasSpatialReferenceSystem

a rdf:Property ;

skos:prefLabel "has spatial reference system"@en ;

.

smls-shacl:hasState

a rdf:Property ;

skos:prefLabel "has state"@en ;

skos:definition "This relationship applies between an Activity and a State or between a Physical Object and a State"@en ;

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

.

smls-shacl:hasTemporalLocation

a rdf:Property ;

skos:prefLabel "has temporal location"@en ;

.

smls-shacl:hasTemporalReference

a rdf:Property ;

skos:prefLabel "has temporal reference"@en ;

.

smls-shacl:hasTemporalReferenceSystem

a rdf:Property ;

skos:prefLabel "has temporal reference system"@en ;

.

smls-shacl:performs

a rdf:Property ;

skos:prefLabel "performs"@en ;

skos:definition "This relationship applies between a Physical Object and an Activity"@en ;

skos:example "The welding of a bridge railing (an activity) is carried out with the aid of a welding machine (physical object) and the welder (physical object)"@en ;

.

smls-shacl:quantityKind

a rdf:Property ;

skos:prefLabel "quantity kind"@en ;

.

smls-shacl:isImplementedBy

a rdf:Property ;

skos:prefLabel "is implemented by"@en ;

skos:definition "This relationship applies between a FunctionalObject and a TechnicalObject. The technical object fullfils or play the role of the functional object"@en ;

.

smls-shacl:isRealizedBy

a rdf:Property ;

skos:prefLabel "is realized by"@en ;

skos:definition "This relationship applies between a PlannedObject and a BuiltObject. The built object corresponds to the planned object"@en ;

.

smls-shacl:quantityKindShape

a sh:NodeShape ;

sh:targetObjectsOf smls-shacl:quantiyKind ;

sh:class qudt:QuantityKind ;

.

smls-shacl:transforms

a rdf:Property ;

skos:prefLabel "transforms"@en ;

skos:definition "This relationship applies between an Activity and a Physical Object or between an Activity and an Information Object"@en ;

skos:example "Baking (an activity) transforms a pizza (a physical object) from unbaked (a state) to baked (another state)"@en ;

.

smls-shacl:triggers

a rdf:Property ;

skos:prefLabel "triggers"@en ;

skos:definition "This relationship applies between an Event and an Activity and an Event and a Physical Object"@en ;

skos:example "Take a traffic accident (an event). This initiates the arrival of the emergency services (an activity)"@en ;

.

smls-shacl:unit

a rdf:Property ;

skos:prefLabel "unit"@en ;

.

smls-shacl:unitShape

a sh:NodeShape ;

sh:targetObjectsOf smls-shacl:unit ;

sh:class qudt:Unit ;

.

1. (normative)  
     
   Selected W3C RDF language subsets
   1. General

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

* 1. XML schema part 2: Datatypes 2nd edition
* xsd:string
* xsd:integer
* xsd:decimal
* xsd:boolean
* xsd:float
* xsd:double
* xsd:anyURI
* xsd:date
* xsd:time
* xsd:dateTime
* xsd:duration
* xsd:minInclusive
* xsd:maxInclusive
* xsd:minExclusive
* xsd:maxExclusive
  1. Resource description framework (RDF)
* rdf:type
* rdf:value
* rdf:HTML
* rdf:Statement
* rdf:subject
* rdf:predicate
* rdf:object
  1. 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
  1. Resource description framework schema (RDFS)
* rdfs:Class
* rdfs:Datatype
* rdfs:subClassOf
* rdfs:subPropertyOf
* rdfs:domain
* rdfs:range
* rdfs:label
* rdfs:comment
* rdfs:isDefinedBy
* rdfs:seeAlso
* rdfs:Container
* rdfs:member
  1. 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
* propertyChainAxiom
* owl:Thing
* owl:topDataProperty
* owl:topObjectProperty
* owl:equivalentClass
* owl:equivalentProperty
* owl:sameAs
  1. Shape constraint language (SHACL)
* sh:NodeShape
* sh:PropertyShape
* sh:property
* sh:path
* sh:inversePath
* sh:targetNode
* sh:targetClass
* sh:targetSubjectsOf
* sh:targetObjectsOf
* sh:nodeKind
* 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

1. (informative)  
     
   Example in SKOS
   1. General

This annex describes typical example code complying to the SML standard. The linked data language used here is SKOS. It is serialised in the three selected alternative concrete syntax forms (RDF/XML, Turtle and JSON-LD).

* 1. Example in SKOS (RDF/XML format)

<?xml version="1.0"?>

<rdf:RDF

    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

    xmlns:smls-skos="https://w3id.org/def/smls-skos#"

    xmlns:ex="https://w3id.org/def/smls-example-skos#"

    xmlns:owl="http://www.w3.org/2002/07/owl#"

    xmlns:skos="http://www.w3.org/2004/02/skos/core#"

    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

  xml:base="https://w3id.org/def/smls-example-skos">

  <skos:ConceptScheme rdf:about="">

    <owl:imports rdf:resource="https://w3id.org/def/smls-skos"/>

    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Ontology"/>

  </skos:ConceptScheme>

  <skos:Concept rdf:ID="Normal">

    <skos:broader>

      <skos:Concept rdf:ID="LoadLevelType"/>

    </skos:broader>

  </skos:Concept>

  <skos:Concept rdf:ID="Deck">

    <skos:broader rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

  </skos:Concept>

  <skos:Concept rdf:ID="Slab">

    <skos:broader rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

  </skos:Concept>

  <skos:Concept rdf:ID="currentlyServingVehicle">

    <skos:related>

      <skos:Concept rdf:ID="Vehicle">

        <skos:broader rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

      </skos:Concept>

    </skos:related>

  </skos:Concept>

  <skos:Concept rdf:ID="height"/>

  <skos:Concept rdf:ID="Bridge">

    <skos:definition xml:lang="nl">Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water.</skos:definition>

    <skos:broader rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

    <skos:prefLabel xml:lang="nl">Brug</skos:prefLabel>

    <skos:prefLabel xml:lang="en">Bridge</skos:prefLabel>

  </skos:Concept>

  <skos:Concept rdf:ID="NonGeometricProperties">

    <skos:narrower>

      <skos:Concept rdf:ID="velocity"/>

    </skos:narrower>

    <skos:narrower>

      <skos:Concept rdf:ID="loadLevel">

        <skos:related rdf:resource="#LoadLevelType"/>

        <skos:related rdf:resource="#Vehicle"/>

      </skos:Concept>

    </skos:narrower>

  </skos:Concept>

  <skos:Concept rdf:ID="Heavy">

    <skos:broader rdf:resource="#LoadLevelType"/>

  </skos:Concept>

  <skos:Concept rdf:ID="Light">

    <skos:broader rdf:resource="#LoadLevelType"/>

  </skos:Concept>

  <skos:Concept rdf:ID="accuracy"/>

  <skos:Concept rdf:ID="hasLegalSpeed">

    <skos:related rdf:resource="#Vehicle"/>

  </skos:Concept>

</rdf:RDF>

* 1. Example in SKOS (Turtle format)

# baseURI: <https://w3id.org/def/smls-example-skos>

# imports: <https://w3id.org/def/smls-skos>

# prefix: ex

@prefix ex: <https://w3id.org/def/smls-example-skos#> .

@prefix smls-skos: <https://w3id.org/def/smls-skos#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<https://w3id.org/def/smls-example-skos>

  a skos:ConceptScheme ;

  owl:imports <https://w3id.org/def/smls-skos> ;

.

ex:Bridge

  a skos:Concept ;

  skos:prefLabel "Bridge"@en ;

  skos:prefLabel "Brug"@nl ;

  skos:broader smls-skos:PhysicalObject ;

  skos:definition "Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water."@nl ;

.

ex:Deck

  a skos:Concept ;

  skos:broader smls-skos:PhysicalObject ;

.

ex:Heavy

  a skos:Concept ;

  skos:broader ex:LoadLevelType ;

.

ex:Light

  a skos:Concept ;

  skos:broader ex:LoadLevelType ;

.

ex:LoadLevelType

  a skos:Concept ;

.

ex:NonGeometricProperties

  a skos:Concept ;

  skos:narrower ex:loadLevel ;

  skos:narrower ex:velocity ;

.

ex:Normal

  a skos:Concept ;

  skos:broader ex:LoadLevelType ;

.

ex:Slab

  a skos:Concept ;

  skos:broader smls-skos:PhysicalObject ;

.

ex:Vehicle

  a skos:Concept ;

  skos:broader smls-skos:PhysicalObject ;

.

ex:accuracy

  a skos:Concept ;

.

ex:currentlyServingVehicle

  a skos:Concept ;

  skos:related ex:Vehicle ;

.

ex:hasLegalSpeed

  a skos:Concept ;

  skos:related ex:Vehicle ;

.

ex:height

  a skos:Concept ;

.

ex:loadLevel

  a skos:Concept ;

  skos:related ex:Vehicle ;

  skos:related ex:LoadLevelType ;

.

ex:velocity

  a skos:Concept ;

.

* 1. Example in SKOS (JSON-LD format)

{

  "@graph" : [ {

    "@id" : "<https://w3id.org/def/smls-example-skos>",

    "@type" : [ "skos:ConceptScheme", "owl:Ontology" ],

    "imports" : "<https://w3id.org/def/smls-skos>"

  }, {

    "@id" : "ex:Bridge",

    "@type" : "skos:Concept",

    "broader" : "smls-skos:PhysicalObject",

    "definition" : {

      "@language" : "nl",

      "@value" : "Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water."

    },

    "prefLabel" : [ {

      "@language" : "nl",

      "@value" : "Brug"

    }, {

      "@language" : "en",

      "@value" : "Bridge"

    } ]

  }, {

    "@id" : "ex:Deck",

    "@type" : "skos:Concept",

    "broader" : "smls-skos:PhysicalObject"

  }, {

    "@id" : "ex:Heavy",

    "@type" : "skos:Concept",

    "broader" : "ex:LoadLevelType"

  }, {

    "@id" : "ex:Light",

    "@type" : "skos:Concept",

    "broader" : "ex:LoadLevelType"

  }, {

    "@id" : "ex:LoadLevelType",

    "@type" : "skos:Concept"

  }, {

    "@id" : "ex:NonGeometricProperties",

    "@type" : "skos:Concept",

    "narrower" : [ "ex:velocity", "ex:loadLevel" ]

  }, {

    "@id" : "ex:Normal",

    "@type" : "skos:Concept",

    "broader" : "ex:LoadLevelType"

  }, {

    "@id" : "ex:Slab",

    "@type" : "skos:Concept",

    "broader" : "smls-skos:PhysicalObject"

  }, {

    "@id" : "ex:Vehicle",

    "@type" : "skos:Concept",

    "broader" : "smls-skos:PhysicalObject"

  }, {

    "@id" : "ex:accuracy",

    "@type" : "skos:Concept"

  }, {

    "@id" : "ex:currentlyServingVehicle",

    "@type" : "skos:Concept",

    "related" : "ex:Vehicle"

  }, {

    "@id" : "ex:hasLegalSpeed",

    "@type" : "skos:Concept",

    "related" : "ex:Vehicle"

  }, {

    "@id" : "ex:height",

    "@type" : "skos:Concept"

  }, {

    "@id" : "ex:loadLevel",

    "@type" : "skos:Concept",

    "related" : [ "ex:LoadLevelType", "ex:Vehicle" ]

  }, {

    "@id" : "ex:velocity",

    "@type" : "skos:Concept"

  } ],

  "@context" : {

    "broader" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#broader>",

      "@type" : "@id"

    },

    "related" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#related>",

      "@type" : "@id"

    },

    "narrower" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#narrower>",

      "@type" : "@id"

    },

    "imports" : {

      "@id" : "<http://www.w3.org/2002/07/owl#imports>",

      "@type" : "@id"

    },

    "definition" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#definition>"

    },

    "prefLabel" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#prefLabel>"

    },

    "ex" : "<https://w3id.org/def/smls-example-skos>#",

    "smls-skos" : "<https://w3id.org/def/smls-skos>#",

    "rdf" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns>#",

    "owl" : "<http://www.w3.org/2002/07/owl>#",

    "xsd" : "<http://www.w3.org/2001/XMLSchema>#",

    "skos" : "<http://www.w3.org/2004/02/skos/core>#",

    "rdfs" : "<http://www.w3.org/2000/01/rdf-schema>#"

  }

}

1. (informative)  
     
   Example in RDFS
   1. General

This annex describes typical example code complying to the SML standard. The linked data language used here is RDFS. It is serialised in the three selected alternative concrete syntax forms (RDF/XML, Turtle and JSON-LD).

* 1. Example in RDFS (RDF/XML format)

<?xml version="1.0"?>

<rdf:RDF

    xmlns:qudt="http://qudt.org/schema/qudt/"

    xmlns:unit="http://qudt.org/vocab/unit/"

    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

    xmlns:owl="http://www.w3.org/2002/07/owl#"

    xmlns:quantitykind="http://qudt.org/vocab/quantitykind/"

    xmlns:smls-rdfs="https://w3id.org/def/smls-owl#"

    xmlns:skos="http://www.w3.org/2004/02/skos/core#"

    xmlns:ex="https://w3id.org/def/smls-example-owl#"

    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

  xml:base="https://w3id.org/def/example">

  <owl:Ontology rdf:about="">

    <owl:imports rdf:resource="https://w3id.org/def/smls-owl"/>

  </owl:Ontology>

  <owl:Class rdf:ID="Deck">

    <rdfs:subClassOf>

      <owl:Restriction>

        <owl:onProperty rdf:resource="https://w3id.org/def/smls-owl#hasPart"/>

        <owl:onClass>

          <owl:Class rdf:ID="Slab"/>

        </owl:onClass>

        <owl:minQualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"

        >1</owl:minQualifiedCardinality>

      </owl:Restriction>

    </rdfs:subClassOf>

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

  </owl:Class>

  <owl:Class rdf:about="#Slab">

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

  </owl:Class>

  <owl:Class rdf:ID="Vehicle">

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

  </owl:Class>

  <owl:Class rdf:ID="LoadLevelType">

    <owl:oneOf rdf:parseType="Collection">

      <ex:LoadLevelType rdf:ID="Light"/>

      <ex:LoadLevelType rdf:ID="Normal"/>

      <ex:LoadLevelType rdf:ID="Heavy"/>

    </owl:oneOf>

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#EnumerationType"/>

  </owl:Class>

  <owl:Class rdf:ID="Bridge">

    <skos:definition xml:lang="nl">Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water.</skos:definition>

    <rdfs:subClassOf>

      <owl:Restriction>

        <owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"

        >1</owl:qualifiedCardinality>

        <owl:onProperty rdf:resource="https://w3id.org/def/smls-owl#hasPart"/>

        <owl:onClass rdf:resource="#Deck"/>

      </owl:Restriction>

    </rdfs:subClassOf>

    <rdfs:subClassOf>

      <owl:Restriction>

        <owl:onProperty>

          <owl:ObjectProperty rdf:ID="height"/>

        </owl:onProperty>

        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"

        >1</owl:cardinality>

      </owl:Restriction>

    </rdfs:subClassOf>

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

    <rdfs:label xml:lang="nl">Brug</rdfs:label>

    <rdfs:label xml:lang="en">Bridge</rdfs:label>

  </owl:Class>

  <owl:ObjectProperty rdf:ID="accuracy">

    <rdfs:range rdf:resource="https://w3id.org/def/smls-owl#QuantityValue"/>

  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="loadLevel">

    <rdfs:domain rdf:resource="#Vehicle"/>

  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="currentlyServingVehicle">

    <rdfs:range rdf:resource="#Vehicle"/>

  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:about="#height">

    <smls-rdfs:quantityKind rdf:resource="http://qudt.org/vocab/quantitykind/Length"/>

    <rdfs:range rdf:resource="https://w3id.org/def/smls-owl#QuantityValue"/>

  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="velocity">

    <smls-rdfs:quantityKind rdf:resource="http://qudt.org/vocab/quantitykind/Speed"/>

    <rdfs:range rdf:resource="https://w3id.org/def/smls-owl#QuantityValue"/>

  </owl:ObjectProperty>

  <owl:DatatypeProperty rdf:ID="hasLegalSpeed">

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>

    <rdfs:domain rdf:resource="#Vehicle"/>

  </owl:DatatypeProperty>

  <rdfs:Container rdf:ID="NonGeometricProperties">

    <rdfs:member rdf:resource="#velocity"/>

    <rdfs:member rdf:resource="#loadLevel"/>

  </rdfs:Container>

  <ex:Bridge rdf:ID="Bridge\_1">

    <ex:height rdf:parseType="Resource">

      <smls-rdfs:unit rdf:resource="http://qudt.org/vocab/unit/M"/>

      <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

      >50.0</rdf:value>

    </ex:height>

    <ex:currentlyServingVehicle>

      <ex:Vehicle rdf:ID="Vehicle\_1">

        <ex:velocity rdf:parseType="Resource">

          <ex:accuracy rdf:parseType="Resource">

            <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

            >0.9</rdf:value>

          </ex:accuracy>

          <smls-rdfs:unit rdf:resource="http://qudt.org/vocab/unit/KiloM-PER-HR"/>

          <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

          >128.0</rdf:value>

        </ex:velocity>

        <ex:loadLevel rdf:resource="#Heavy"/>

        <ex:hasLegalSpeed rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"

        >true</ex:hasLegalSpeed>

      </ex:Vehicle>

    </ex:currentlyServingVehicle>

    <smls-rdfs:hasPart>

      <ex:Deck rdf:ID="Deck\_1">

        <smls-rdfs:hasPart>

          <ex:Slab rdf:ID="Slab\_3"/>

        </smls-rdfs:hasPart>

        <smls-rdfs:hasPart>

          <ex:Slab rdf:ID="Slab\_2"/>

        </smls-rdfs:hasPart>

        <smls-rdfs:hasPart>

          <ex:Slab rdf:ID="Slab\_1"/>

        </smls-rdfs:hasPart>

      </ex:Deck>

    </smls-rdfs:hasPart>

  </ex:Bridge>

</rdf:RDF>

* 1. Example in RDFS (Turtle format)

# baseURI: https://w3id.org/def/smls-example-owl

# imports: https://w3id.org/def/smls-owl

# prefix: ex

@prefix smls-rdfs: <https://w3id.org/def/smls-owl#> .

@prefix ex: <https://w3id.org/def/smls-example-owl#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .

@prefix qudt: <http://qudt.org/schema/qudt/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix unit: <http://qudt.org/vocab/unit/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/def/smls-example-owl>

  a owl:Ontology ;

  owl:imports <https://w3id.org/def/smls-owl> ;

.

ex:Bridge

  a owl:Class ;

  rdfs:label "Bridge"@en ;

  rdfs:label "Brug"@nl ;

  rdfs:subClassOf smls-rdfs:PhysicalObject ;

  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 smls-rdfs:hasPart ;

      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;

    ] ;

  skos:definition "Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water."@nl ;

.

ex:Bridge\_1

  a ex:Bridge ;

  smls-rdfs:hasPart ex:Deck\_1 ;

  ex:currentlyServingVehicle ex:Vehicle\_1 ;

  ex:height [

      rdf:value 50.0 ;

      smls-rdfs:unit unit:M ;

    ] ;

.

ex:Deck

  a owl:Class ;

  rdfs:subClassOf smls-rdfs:PhysicalObject ;

  rdfs:subClassOf [

      a owl:Restriction ;

      owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;

      owl:onClass ex:Slab ;

      owl:onProperty smls-rdfs:hasPart ;

    ] ;

.

ex:Deck\_1

  a ex:Deck ;

  smls-rdfs:hasPart ex:Slab\_1 ;

  smls-rdfs:hasPart ex:Slab\_2 ;

  smls-rdfs:hasPart ex:Slab\_3 ;

.

ex:Heavy

  a ex:LoadLevelType ;

.

ex:Light

  a ex:LoadLevelType ;

.

ex:LoadLevelType

  a owl:Class ;

  rdfs:subClassOf smls-rdfs:EnumerationType ;

  owl:oneOf (

      ex:Light

      ex:Normal

      ex:Heavy

    ) ;

.

ex:NonGeometricProperties

  a rdfs:Container ;

  rdfs:member ex:loadLevel ;

  rdfs:member ex:velocity ;

.

ex:Normal

  a ex:LoadLevelType ;

.

ex:Slab

  a owl:Class ;

  rdfs:subClassOf smls-rdfs:PhysicalObject ;

.

ex:Slab\_1

  a ex:Slab ;

.

ex:Slab\_2

  a ex:Slab ;

.

ex:Slab\_3

  a ex:Slab ;

.

ex:Vehicle

  a owl:Class ;

  rdfs:subClassOf smls-rdfs:PhysicalObject ;

.

ex:Vehicle\_1

  a ex:Vehicle ;

  ex:hasLegalSpeed true ;

  ex:loadLevel ex:Heavy ;

  ex:velocity [

      rdf:value 128.0 ;

      smls-rdfs:unit unit:KiloM-PER-HR ;

      ex:accuracy [

          rdf:value 0.9 ;

        ] ;

    ] ;

.

ex:accuracy

  a owl:ObjectProperty ;

  rdfs:range smls-rdfs:QuantityValue ;

.

ex:currentlyServingVehicle

  a owl:ObjectProperty ;

  rdfs:range ex:Vehicle ;

.

ex:hasLegalSpeed

  a owl:DatatypeProperty ;

  rdfs:domain ex:Vehicle ;

  rdfs:range xsd:boolean ;

.

ex:height

  a owl:ObjectProperty ;

  rdfs:range smls-rdfs:QuantityValue ;

  smls-rdfs:quantityKind quantitykind:Length ;

.

ex:loadLevel

  a owl:ObjectProperty ;

  rdfs:domain ex:Vehicle ;

  rdfs:range ex:LoadLevelType ;

.

ex:velocity

  a owl:ObjectProperty ;

  rdfs:range smls-rdfs:QuantityValue ;

  smls-rdfs:quantityKind quantitykind:Speed ;

.

* 1. Example in RDFS (JSON-LD format)

{

  "@graph" : [ {

    "@id" : "\_:b0",

    "value" : "50.0",

    "smls-rdfs:unit" : {

      "@id" : "unit:M"

    }

  }, {

    "@id" : "\_:b3",

    "value" : "128.0",

    "accuracy" : "\_:b4",

    "smls-rdfs:unit" : {

      "@id" : "unit:KiloM-PER-HR"

    }

  }, {

    "@id" : "\_:b4",

    "value" : "0.9"

  }, {

    "@id" : "<https://w3id.org/def/smls-example-rdfs>",

    "@type" : "owl:Ontology",

    "imports" : "<https://w3id.org/def/smls-rdfs>"

  }, {

    "@id" : "ex:Bridge",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-rdfs:PhysicalObject",

    "definition" : {

      "@language" : "nl",

      "@value" : "Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water."

    },

    "prefLabel" : [ {

      "@language" : "nl",

      "@value" : "Brug"

    }, {

      "@language" : "en",

      "@value" : "Bridge"

    } ]

  }, {

    "@id" : "ex:Bridge\_1",

    "rdf:type" : {

      "@id" : "ex:Bridge"

    },

    "currentlyServingVehicle" : "ex:Vehicle\_1",

    "height" : "\_:b0",

    "hasPart" : "ex:Deck\_1"

  }, {

    "@id" : "ex:Deck",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-rdfs:PhysicalObject"

  }, {

    "@id" : "ex:Deck\_1",

    "rdf:type" : {

      "@id" : "ex:Deck"

    },

    "hasPart" : [ "ex:Slab\_3", "ex:Slab\_2", "ex:Slab\_1" ]

  }, {

    "@id" : "ex:Heavy",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:Light",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:LoadLevelType",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-rdfs:EnumerationType",

    "oneOf" : {

      "@list" : [ "ex:Light", "ex:Normal", "ex:Heavy" ]

    }

  }, {

    "@id" : "ex:NonGeometricProperties",

    "@type" : "rdfs:Container",

    "member" : [ "ex:velocity", "ex:loadLevel" ]

  }, {

    "@id" : "ex:Normal",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:Slab",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-rdfs:PhysicalObject"

  }, {

    "@id" : "ex:Slab\_1",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Slab\_2",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Slab\_3",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Vehicle",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-rdfs:PhysicalObject"

  }, {

    "@id" : "ex:Vehicle\_1",

    "rdf:type" : {

      "@id" : "ex:Vehicle"

    },

    "ex:hasLegalSpeed" : true,

    "loadLevel" : "ex:Heavy",

    "velocity" : "\_:b3"

  }, {

    "@id" : "ex:accuracy",

    "@type" : "rdf:Property",

    "range" : "smls-rdfs:QuantityValue"

  }, {

    "@id" : "ex:currentlyServingVehicle",

    "@type" : "rdf:Property",

    "range" : "ex:Vehicle"

  }, {

    "@id" : "ex:hasLegalSpeed",

    "@type" : "rdf:Property",

    "domain" : "ex:Vehicle",

    "range" : "xsd:boolean"

  }, {

    "@id" : "ex:height",

    "@type" : "rdf:Property",

    "range" : "smls-rdfs:QuantityValue",

    "quantityKind" : "quantitykind:Length"

  }, {

    "@id" : "ex:loadLevel",

    "@type" : "rdf:Property",

    "domain" : "ex:Vehicle",

    "range" : "ex:LoadLevelType"

  }, {

    "@id" : "ex:velocity",

    "@type" : "rdf:Property",

    "range" : "smls-rdfs:QuantityValue",

    "quantityKind" : "quantitykind:Speed"

  } ],

  "@context" : {

    "subClassOf" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#subClassOf>",

      "@type" : "@id"

    },

    "unit" : "<http://qudt.org/vocab/unit/>",

    "value" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#value>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#decimal>"

    },

    "range" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#range>",

      "@type" : "@id"

    },

    "quantityKind" : {

      "@id" : "<https://w3id.org/def/smls-rdfs#quantityKind>",

      "@type" : "@id"

    },

    "rest" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>",

      "@type" : "@id"

    },

    "first" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#first>",

      "@type" : "@id"

    },

    "accuracy" : {

      "@id" : "<https://w3id.org/def/smls-example-rdfs#accuracy>",

      "@type" : "@id"

    },

    "member" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#member>",

      "@type" : "@id"

    },

    "oneOf" : {

      "@id" : "<http://www.w3.org/2002/07/owl#oneOf>",

      "@type" : "@id"

    },

    "height" : {

      "@id" : "<https://w3id.org/def/smls-example-rdfs#height>",

      "@type" : "@id"

    },

    "currentlyServingVehicle" : {

      "@id" : "<https://w3id.org/def/smls-example-rdfs#currentlyServingVehicle>",

      "@type" : "@id"

    },

    "hasPart" : {

      "@id" : "<https://w3id.org/def/smls-rdfs#hasPart>",

      "@type" : "@id"

    },

    "domain" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#domain>",

      "@type" : "@id"

    },

    "velocity" : {

      "@id" : "<https://w3id.org/def/smls-example-rdfs#velocity>",

      "@type" : "@id"

    },

    "loadLevel" : {

      "@id" : "<https://w3id.org/def/smls-example-rdfs#loadLevel>",

      "@type" : "@id"

    },

    "hasLegalSpeed" : {

      "@id" : "<https://w3id.org/def/smls-example-rdfs#hasLegalSpeed>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#boolean>"

    },

    "imports" : {

      "@id" : "<http://www.w3.org/2002/07/owl#imports>",

      "@type" : "@id"

    },

    "definition" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#definition>"

    },

    "prefLabel" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#prefLabel>"

    },

    "quantitykind" : "<http://qudt.org/vocab/quantitykind/>",

    "qudt" : "<http://qudt.org/schema/qudt/>",

    "ex" : "<https://w3id.org/def/smls-example-rdfs>#",

    "smls-rdfs" : "<https://w3id.org/def/smls-rdfs>#",

    "rdf" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns>#",

    "owl" : "<http://www.w3.org/2002/07/owl>#",

    "xsd" : "<http://www.w3.org/2001/XMLSchema>#",

    "skos" : "<http://www.w3.org/2004/02/skos/core>#",

    "rdfs" : "<http://www.w3.org/2000/01/rdf-schema>#"

  }

}

1. (informative)  
     
   Example in OWL
   1. General

This annex describes typical example code complying to the SML standard. The linked data language used here is OWL. It is serialised in the three selected alternative concrete syntax forms (RDF/XML, Turtle and JSON-LD).

* 1. Example in OWL (RDF/XML format)

<?xml version="1.0"?>

<rdf:RDF

    xmlns:qudt="http://qudt.org/schema/qudt/"

    xmlns:unit="http://qudt.org/vocab/unit/"

    xmlns:ex="https://w3id.org/def/smls-example-owl#"

    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

    xmlns:smls-owl="https://w3id.org/def/smls-owl#"

    xmlns:owl="http://www.w3.org/2002/07/owl#"

    xmlns:quantitykind="http://qudt.org/vocab/quantitykind/"

    xmlns:skos="http://www.w3.org/2004/02/skos/core#"

    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

  xml:base="https://w3id.org/def/smls-example-owl">

  <owl:Ontology rdf:about="">

    <owl:imports rdf:resource="https://w3id.org/def/smls-owl"/>

  </owl:Ontology>

  <owl:Class rdf:ID="Slab">

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

  </owl:Class>

  <owl:Class rdf:ID="Bridge">

    <skos:definition xml:lang="nl">Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water.</skos:definition>

    <rdfs:subClassOf>

      <owl:Restriction>

        <owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"

        >1</owl:qualifiedCardinality>

        <owl:onProperty rdf:resource="https://w3id.org/def/smls-owl#hasPart"/>

        <owl:onClass>

          <owl:Class rdf:ID="Deck"/>

        </owl:onClass>

      </owl:Restriction>

    </rdfs:subClassOf>

    <rdfs:subClassOf>

      <owl:Restriction>

        <owl:onProperty>

          <owl:ObjectProperty rdf:ID="height"/>

        </owl:onProperty>

        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"

        >1</owl:cardinality>

      </owl:Restriction>

    </rdfs:subClassOf>

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

    <skos:prefLabel xml:lang="nl">Brug</skos:prefLabel>

    <skos:prefLabel xml:lang="en">Bridge</skos:prefLabel>

  </owl:Class>

  <owl:Class rdf:ID="LoadLevelType">

    <owl:oneOf rdf:parseType="Collection">

      <ex:LoadLevelType rdf:ID="Light"/>

      <ex:LoadLevelType rdf:ID="Normal"/>

      <ex:LoadLevelType rdf:ID="Heavy"/>

    </owl:oneOf>

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#EnumerationType"/>

  </owl:Class>

  <owl:Class rdf:about="#Deck">

    <rdfs:subClassOf>

      <owl:Restriction>

        <owl:onProperty rdf:resource="https://w3id.org/def/smls-owl#hasPart"/>

        <owl:onClass rdf:resource="#Slab"/>

        <owl:minQualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"

        >1</owl:minQualifiedCardinality>

      </owl:Restriction>

    </rdfs:subClassOf>

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

  </owl:Class>

  <owl:Class rdf:ID="Vehicle">

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-owl#PhysicalObject"/>

  </owl:Class>

  <owl:ObjectProperty rdf:ID="velocity">

    <smls-owl:quantityKind rdf:resource="http://qudt.org/vocab/quantitykind/Speed"/>

    <rdfs:range rdf:resource="https://w3id.org/def/smls-owl#QuantityValue"/>

  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="loadLevel">

    <rdfs:range rdf:resource="#LoadLevelType"/>

    <rdfs:domain rdf:resource="#Vehicle"/>

  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:about="#height">

    <smls-owl:quantityKind rdf:resource="http://qudt.org/vocab/quantitykind/Length"/>

    <rdfs:range rdf:resource="https://w3id.org/def/smls-owl#QuantityValue"/>

  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="accuracy">

    <rdfs:range rdf:resource="https://w3id.org/def/smls-owl#QuantityValue"/>

  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="currentlyServingVehicle">

    <rdfs:range rdf:resource="#Vehicle"/>

  </owl:ObjectProperty>

  <owl:DatatypeProperty rdf:ID="hasLegalSpeed">

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>

    <rdfs:domain rdf:resource="#Vehicle"/>

  </owl:DatatypeProperty>

  <ex:Bridge rdf:ID="Bridge\_1">

    <ex:height rdf:parseType="Resource">

      <smls-owl:unit rdf:resource="http://qudt.org/vocab/unit/M"/>

      <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

      >50.0</rdf:value>

    </ex:height>

    <ex:currentlyServingVehicle>

      <ex:Vehicle rdf:ID="Vehicle\_1">

        <ex:velocity rdf:parseType="Resource">

          <ex:accuracy rdf:parseType="Resource">

            <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

            >0.9</rdf:value>

          </ex:accuracy>

          <smls-owl:unit rdf:resource="http://qudt.org/vocab/unit/KiloM-PER-HR"/>

          <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

          >128.0</rdf:value>

        </ex:velocity>

        <ex:loadLevel rdf:resource="#Heavy"/>

        <ex:hasLegalSpeed rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"

        >true</ex:hasLegalSpeed>

      </ex:Vehicle>

    </ex:currentlyServingVehicle>

    <smls-owl:hasPart>

      <ex:Deck rdf:ID="Deck\_1">

        <smls-owl:hasPart>

          <ex:Slab rdf:ID="Slab\_3"/>

        </smls-owl:hasPart>

        <smls-owl:hasPart>

          <ex:Slab rdf:ID="Slab\_2"/>

        </smls-owl:hasPart>

        <smls-owl:hasPart>

          <ex:Slab rdf:ID="Slab\_1"/>

        </smls-owl:hasPart>

      </ex:Deck>

    </smls-owl:hasPart>

  </ex:Bridge>

  <rdfs:Container rdf:ID="NonGeometricProperties">

    <rdfs:member rdf:resource="#velocity"/>

    <rdfs:member rdf:resource="#loadLevel"/>

  </rdfs:Container>

</rdf:RDF>

* 1. Example in OWL (Turtle format)

# baseURI: https://w3id.org/def/smls-example-owl

# imports: https://w3id.org/def/smls-owl

# prefix: ex

@prefix smls-owl: <https://w3id.org/def/smls-owl#> .

@prefix ex: <https://w3id.org/def/smls-example-owl#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .

@prefix qudt: <http://qudt.org/schema/qudt/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix unit: <http://qudt.org/vocab/unit/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/def/smls-example-owl>

  a owl:Ontology ;

  owl:imports <https://w3id.org/def/smls-owl> ;

.

ex:Bridge

  a owl:Class ;

  skos:prefLabel "Bridge"@en ;

  skos:prefLabel "Brug"@nl ;

  rdfs:subClassOf smls-owl:PhysicalObject ;

  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 smls-owl:hasPart ;

      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;

    ] ;

  skos:definition "Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water."@nl ;

.

ex:Bridge\_1

  a ex:Bridge ;

  smls-owl:hasPart ex:Deck\_1 ;

  ex:currentlyServingVehicle ex:Vehicle\_1 ;

  ex:height [

      rdf:value 50.0 ;

      smls-owl:unit unit:M ;

    ] ;

.

ex:Deck

  a owl:Class ;

  rdfs:subClassOf smls-owl:PhysicalObject ;

  rdfs:subClassOf [

      a owl:Restriction ;

      owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;

      owl:onClass ex:Slab ;

      owl:onProperty smls-owl:hasPart ;

    ] ;

.

ex:Deck\_1

  a ex:Deck ;

  smls-owl:hasPart ex:Slab\_1 ;

  smls-owl:hasPart ex:Slab\_2 ;

  smls-owl:hasPart ex:Slab\_3 ;

.

ex:Heavy

  a ex:LoadLevelType ;

.

ex:Light

  a ex:LoadLevelType ;

.

ex:LoadLevelType

  a owl:Class ;

  rdfs:subClassOf smls-owl:EnumerationType ;

  owl:oneOf (

      ex:Light

      ex:Normal

      ex:Heavy

    ) ;

.

ex:NonGeometricProperties

  a rdfs:Container ;

  rdfs:member ex:loadLevel ;

  rdfs:member ex:velocity ;

.

ex:Normal

  a ex:LoadLevelType ;

.

ex:Slab

  a owl:Class ;

  rdfs:subClassOf smls-owl:PhysicalObject ;

.

ex:Slab\_1

  a ex:Slab ;

.

ex:Slab\_2

  a ex:Slab ;

.

ex:Slab\_3

  a ex:Slab ;

.

ex:Vehicle

  a owl:Class ;

  rdfs:subClassOf smls-owl:PhysicalObject ;

.

ex:Vehicle\_1

  a ex:Vehicle ;

  ex:hasLegalSpeed true ;

  ex:loadLevel ex:Heavy ;

  ex:velocity [

      rdf:value 128.0 ;

      smls-owl:unit unit:KiloM-PER-HR ;

      ex:accuracy [

          rdf:value 0.9 ;

        ] ;

    ] ;

.

ex:accuracy

  a owl:ObjectProperty ;

  rdfs:range smls-owl:QuantityValue ;

.

ex:currentlyServingVehicle

  a owl:ObjectProperty ;

  rdfs:range ex:Vehicle ;

.

ex:hasLegalSpeed

  a owl:DatatypeProperty ;

  rdfs:domain ex:Vehicle ;

  rdfs:range xsd:boolean ;

.

ex:height

  a owl:ObjectProperty ;

  rdfs:range smls-owl:QuantityValue ;

  smls-owl:quantityKind quantitykind:Length ;

.

ex:loadLevel

  a owl:ObjectProperty ;

  rdfs:domain ex:Vehicle ;

  rdfs:domain ex:LoadLevelType ;

.

ex:velocity

  a owl:ObjectProperty ;

  rdfs:range smls-owl:QuantityValue ;

  smls-owl:quantityKind quantitykind:Speed ;

.

* 1. Example in OWL (JSON-LD format)

{

  "@graph" : [ {

    "@id" : "\_:b2",

    "@type" : "owl:Restriction",

    "onClass" : "ex:Deck",

    "onProperty" : "smls-owl:hasPart",

    "qualifiedCardinality" : "1"

  }, {

    "@id" : "\_:b3",

    "@type" : "owl:Restriction",

    "cardinality" : "1",

    "onProperty" : "ex:height"

  }, {

    "@id" : "\_:b4",

    "@type" : "owl:Restriction",

    "minQualifiedCardinality" : "1",

    "onClass" : "ex:Slab",

    "onProperty" : "smls-owl:hasPart"

  }, {

    "@id" : "\_:b5",

    "value" : "0.9"

  }, {

    "@id" : "\_:b7",

    "value" : "128.0",

    "accuracy" : "\_:b5",

    "smls-owl:unit" : {

      "@id" : "unit:KiloM-PER-HR"

    }

  }, {

    "@id" : "\_:b8",

    "value" : "50.0",

    "smls-owl:unit" : {

      "@id" : "unit:M"

    }

  }, {

    "@id" : "<https://w3id.org/def/smls-example-owl>",

    "@type" : "owl:Ontology",

    "imports" : "<https://w3id.org/def/smls-owl>"

  }, {

    "@id" : "ex:Bridge",

    "@type" : "owl:Class",

    "subClassOf" : [ "\_:b2", "\_:b3", "smls-owl:PhysicalObject" ],

    "definition" : {

      "@language" : "nl",

      "@value" : "Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water."

    },

    "prefLabel" : [ {

      "@language" : "nl",

      "@value" : "Brug"

    }, {

      "@language" : "en",

      "@value" : "Bridge"

    } ]

  }, {

    "@id" : "ex:Bridge\_1",

    "rdf:type" : {

      "@id" : "ex:Bridge"

    },

    "currentlyServingVehicle" : "ex:Vehicle\_1",

    "height" : "\_:b8",

    "hasPart" : "ex:Deck\_1"

  }, {

    "@id" : "ex:Deck",

    "@type" : "owl:Class",

    "subClassOf" : [ "\_:b4", "smls-owl:PhysicalObject" ]

  }, {

    "@id" : "ex:Deck\_1",

    "rdf:type" : {

      "@id" : "ex:Deck"

    },

    "hasPart" : [ "ex:Slab\_3", "ex:Slab\_2", "ex:Slab\_1" ]

  }, {

    "@id" : "ex:Heavy",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:Light",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:LoadLevelType",

    "@type" : "owl:Class",

    "subClassOf" : "smls-owl:EnumerationType",

    "oneOf" : {

      "@list" : [ "ex:Light", "ex:Normal", "ex:Heavy" ]

    }

  }, {

    "@id" : "ex:NonGeometricProperties",

    "@type" : "rdfs:Container",

    "member" : [ "ex:velocity", "ex:loadLevel" ]

  }, {

    "@id" : "ex:Normal",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:Slab",

    "@type" : "owl:Class",

    "subClassOf" : "smls-owl:PhysicalObject"

  }, {

    "@id" : "ex:Slab\_1",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Slab\_2",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Slab\_3",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Vehicle",

    "@type" : "owl:Class",

    "subClassOf" : "smls-owl:PhysicalObject"

  }, {

    "@id" : "ex:Vehicle\_1",

    "rdf:type" : {

      "@id" : "ex:Vehicle"

    },

    "ex:hasLegalSpeed" : true,

    "loadLevel" : "ex:Heavy",

    "velocity" : "\_:b7"

  }, {

    "@id" : "ex:accuracy",

    "@type" : "owl:ObjectProperty",

    "range" : "smls-owl:QuantityValue"

  }, {

    "@id" : "ex:currentlyServingVehicle",

    "@type" : "owl:ObjectProperty",

    "range" : "ex:Vehicle"

  }, {

    "@id" : "ex:hasLegalSpeed",

    "@type" : "owl:DatatypeProperty",

    "domain" : "ex:Vehicle",

    "range" : "xsd:boolean"

  }, {

    "@id" : "ex:height",

    "@type" : "owl:ObjectProperty",

    "range" : "smls-owl:QuantityValue",

    "quantityKind" : "quantitykind:Length"

  }, {

    "@id" : "ex:loadLevel",

    "@type" : "owl:ObjectProperty",

    "domain" : "ex:Vehicle",

    "range" : "ex:LoadLevelType"

  }, {

    "@id" : "ex:velocity",

    "@type" : "owl:ObjectProperty",

    "range" : "smls-owl:QuantityValue",

    "quantityKind" : "quantitykind:Speed"

  } ],

  "@context" : {

    "rest" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>",

      "@type" : "@id"

    },

    "first" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#first>",

      "@type" : "@id"

    },

    "quantityKind" : {

      "@id" : "<https://w3id.org/def/smls-owl#quantityKind>",

      "@type" : "@id"

    },

    "range" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#range>",

      "@type" : "@id"

    },

    "subClassOf" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#subClassOf>",

      "@type" : "@id"

    },

    "definition" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#definition>"

    },

    "prefLabel" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#prefLabel>"

    },

    "onProperty" : {

      "@id" : "<http://www.w3.org/2002/07/owl#onProperty>",

      "@type" : "@id"

    },

    "onClass" : {

      "@id" : "<http://www.w3.org/2002/07/owl#onClass>",

      "@type" : "@id"

    },

    "minQualifiedCardinality" : {

      "@id" : "<http://www.w3.org/2002/07/owl#minQualifiedCardinality>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#nonNegativeInteger>"

    },

    "value" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#value>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#decimal>"

    },

    "domain" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#domain>",

      "@type" : "@id"

    },

    "oneOf" : {

      "@id" : "<http://www.w3.org/2002/07/owl#oneOf>",

      "@type" : "@id"

    },

    "member" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#member>",

      "@type" : "@id"

    },

    "hasPart" : {

      "@id" : "<https://w3id.org/def/smls-owl#hasPart>",

      "@type" : "@id"

    },

    "qualifiedCardinality" : {

      "@id" : "<http://www.w3.org/2002/07/owl#qualifiedCardinality>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#nonNegativeInteger>"

    },

    "velocity" : {

      "@id" : "<https://w3id.org/def/smls-example-owl#velocity>",

      "@type" : "@id"

    },

    "loadLevel" : {

      "@id" : "<https://w3id.org/def/smls-example-owl#loadLevel>",

      "@type" : "@id"

    },

    "hasLegalSpeed" : {

      "@id" : "<https://w3id.org/def/smls-example-owl#hasLegalSpeed>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#boolean>"

    },

    "accuracy" : {

      "@id" : "<https://w3id.org/def/smls-example-owl#accuracy>",

      "@type" : "@id"

    },

    "unit" : "<http://qudt.org/vocab/unit/>",

    "imports" : {

      "@id" : "<http://www.w3.org/2002/07/owl#imports>",

      "@type" : "@id"

    },

    "cardinality" : {

      "@id" : "<http://www.w3.org/2002/07/owl#cardinality>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#nonNegativeInteger>"

    },

    "height" : {

      "@id" : "<https://w3id.org/def/smls-example-owl#height>",

      "@type" : "@id"

    },

    "currentlyServingVehicle" : {

      "@id" : "<https://w3id.org/def/smls-example-owl#currentlyServingVehicle>",

      "@type" : "@id"

    },

    "quantitykind" : "<http://qudt.org/vocab/quantitykind/>",

    "qudt" : "<http://qudt.org/schema/qudt/>",

    "ex" : "<https://w3id.org/def/smls-example-owl>#",

    "smls-owl" : "<https://w3id.org/def/smls-owl>#",

    "rdf" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns>#",

    "owl" : "<http://www.w3.org/2002/07/owl>#",

    "xsd" : "<http://www.w3.org/2001/XMLSchema>#",

    "skos" : "<http://www.w3.org/2004/02/skos/core>#",

    "rdfs" : "<http://www.w3.org/2000/01/rdf-schema>#"

  }

}

1. (informative)  
     
   Example in SHACL
   1. General

This annex describes typical example code complying to the SML standard. The linked data language used here is SHACL. It is serialised in the three selected alternative concrete syntax forms (RDF/XML, Turtle and JSON-LD).

* 1. Example in SHACL (RDF/XML format)

<?xml version="1.0"?>

<rdf:RDF

    xmlns:qudt="http://qudt.org/schema/qudt/"

    xmlns:unit="http://qudt.org/vocab/unit/"

    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

    xmlns:ex="https://w3id.org/def/smls-example-shacl#"

    xmlns:owl="http://www.w3.org/2002/07/owl#"

    xmlns:quantitykind="http://qudt.org/vocab/quantitykind/"

    xmlns:smls-shacl="https://w3id.org/def/smls-shacl#"

    xmlns:skos="http://www.w3.org/2004/02/skos/core#"

    xmlns:sh="http://www.w3.org/ns/shacl#"

    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

    xmlns:dash="http://datashapes.org/dash#"

    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

  xml:base="https://w3id.org/def/smls-example-shacl">

  <owl:Ontology rdf:about="">

    <owl:imports rdf:resource="https://w3id.org/def/smls-shacl"/>

  </owl:Ontology>

  <rdfs:Class rdf:ID="Slab">

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-shacl#PhysicalObject"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="Bridge">

    <skos:definition xml:lang="nl">Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water.</skos:definition>

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-shacl#PhysicalObject"/>

    <skos:prefLabel xml:lang="nl">Brug</skos:prefLabel>

    <skos:prefLabel xml:lang="en">Bridge</skos:prefLabel>

  </rdfs:Class>

  <rdfs:Class rdf:ID="Deck">

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-shacl#PhysicalObject"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="Vehicle">

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-shacl#PhysicalObject"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="LoadLevelType">

    <rdfs:subClassOf rdf:resource="https://w3id.org/def/smls-shacl#EnumerationType"/>

  </rdfs:Class>

  <rdf:Property rdf:ID="currentlyServingVehicle"/>

  <rdf:Property rdf:ID="accuracy"/>

  <rdf:Property rdf:ID="loadLevel"/>

  <rdf:Property rdf:ID="height">

    <smls-shacl:quantityKind rdf:resource="http://qudt.org/vocab/quantitykind/Length"/>

  </rdf:Property>

  <rdf:Property rdf:ID="velocity">

    <smls-shacl:quantityKind rdf:resource="http://qudt.org/vocab/quantitykind/Speed"/>

  </rdf:Property>

  <rdf:Property rdf:ID="hasLegalSpeed"/>

  <ex:Bridge rdf:ID="Bridge\_1">

    <ex:height rdf:parseType="Resource">

      <smls-shacl:unit rdf:resource="http://qudt.org/vocab/unit/M"/>

      <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

      >50.0</rdf:value>

    </ex:height>

    <ex:currentlyServingVehicle>

      <ex:Vehicle rdf:ID="Vehicle\_1">

        <ex:velocity rdf:parseType="Resource">

          <ex:accuracy rdf:parseType="Resource">

            <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

            >0.9</rdf:value>

          </ex:accuracy>

          <smls-shacl:unit rdf:resource="http://qudt.org/vocab/unit/KiloM-PER-HR"/>

          <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"

          >128.0</rdf:value>

        </ex:velocity>

        <ex:loadLevel>

          <ex:LoadLevelType rdf:ID="Heavy"/>

        </ex:loadLevel>

        <ex:hasLegalSpeed rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"

        >true</ex:hasLegalSpeed>

      </ex:Vehicle>

    </ex:currentlyServingVehicle>

    <smls-shacl:hasPart>

      <ex:Deck rdf:ID="Deck\_1">

        <smls-shacl:hasPart>

          <ex:Slab rdf:ID="Slab\_3"/>

        </smls-shacl:hasPart>

        <smls-shacl:hasPart>

          <ex:Slab rdf:ID="Slab\_2"/>

        </smls-shacl:hasPart>

        <smls-shacl:hasPart>

          <ex:Slab rdf:ID="Slab\_1"/>

        </smls-shacl:hasPart>

      </ex:Deck>

    </smls-shacl:hasPart>

  </ex:Bridge>

  <sh:NodeShape rdf:ID="currentlyServingVehicleShape">

    <sh:class rdf:resource="https://w3id.org/def/smls-shacl#Vehicle"/>

    <sh:targetObjectsOf rdf:resource="#currentlyServingVehicle"/>

  </sh:NodeShape>

  <sh:NodeShape rdf:ID="DeckShape">

    <sh:property rdf:parseType="Resource">

      <sh:class rdf:resource="#Slab"/>

      <sh:minCount rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"

      >1</sh:minCount>

      <sh:path rdf:resource="https://w3id.org/def/smls-shacl#hasPart"/>

    </sh:property>

    <sh:targetClass rdf:resource="#Deck"/>

  </sh:NodeShape>

  <sh:NodeShape rdf:ID="VehicleShape">

    <sh:property rdf:parseType="Resource">

      <sh:datatype rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>

      <sh:path rdf:resource="#hasLegalSpeed"/>

    </sh:property>

    <sh:property rdf:parseType="Resource">

      <sh:in rdf:parseType="Collection">

        <ex:LoadLevelType rdf:ID="Light"/>

        <ex:LoadLevelType rdf:ID="Normal"/>

        <ex:LoadLevelType rdf:about="#Heavy"/>

      </sh:in>

      <sh:class rdf:resource="#LoadLevelType"/>

      <sh:path rdf:resource="#loadLevel"/>

    </sh:property>

    <sh:targetClass rdf:resource="#Vehicle"/>

  </sh:NodeShape>

  <rdfs:Container rdf:ID="NonGeometricProperties">

    <rdfs:member rdf:resource="#velocity"/>

    <rdfs:member rdf:resource="#loadLevel"/>

  </rdfs:Container>

  <sh:NodeShape rdf:ID="velocityShape">

    <sh:class rdf:resource="https://w3id.org/def/smls-shacl#QuantityValue"/>

    <sh:targetObjectsOf rdf:resource="#velocity"/>

  </sh:NodeShape>

  <sh:NodeShape rdf:ID="heightShape">

    <sh:class rdf:resource="https://w3id.org/def/smls-shacl#QuantityValue"/>

    <sh:targetObjectsOf rdf:resource="#height"/>

  </sh:NodeShape>

  <sh:NodeShape rdf:ID="accuracyShape">

    <sh:class rdf:resource="https://w3id.org/def/smls-shacl#QuantityValue"/>

    <sh:targetObjectsOf rdf:resource="#accuracy"/>

  </sh:NodeShape>

  <sh:NodeShape rdf:ID="BridgeShape">

    <sh:property rdf:parseType="Resource">

      <sh:class rdf:resource="#Deck"/>

      <sh:maxCount rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"

      >1</sh:maxCount>

      <sh:minCount rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"

      >1</sh:minCount>

      <sh:path rdf:resource="https://w3id.org/def/smls-shacl#hasPart"/>

    </sh:property>

    <sh:property rdf:parseType="Resource">

      <sh:minCount rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"

      >1</sh:minCount>

      <sh:path rdf:resource="#height"/>

    </sh:property>

    <sh:targetClass rdf:resource="#Bridge"/>

  </sh:NodeShape>

</rdf:RDF>

* 1. Example in SHACL (Turtle format)

# baseURI: <https://w3id.org/def/smls-example-shacl>

# imports: <https://w3id.org/def/smls-shacl>

# prefix: ex

@prefix smls-shacl: <https://w3id.org/def/smls-shacl#> .

@prefix ex: <https://w3id.org/def/smls-example-shacl#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> .

@prefix qudt: <http://qudt.org/schema/qudt/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix unit: <http://qudt.org/vocab/unit/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix dash: <http://datashapes.org/dash#> .

@prefix sh: <http://www.w3.org/ns/shacl#> .

<https://w3id.org/def/smls-example-shacl>

  a owl:Ontology ;

  owl:imports <https://w3id.org/def/smls-shacl> ;

.

ex:Bridge

  a rdfs:Class ;

  skos:prefLabel "Bridge"@en ;

  skos:prefLabel "Brug"@nl ;

  rdfs:subClassOf smls-shacl:PhysicalObject ;

  skos:definition "Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water."@nl ;

.

ex:BridgeShape

  a sh:NodeShape ;

  sh:targetClass ex:Bridge ;

  sh:property [

    sh:path ex:height ;

    sh:minCount 1 ;

    ] ;

  sh:property [

    sh:path smls-shacl:hasPart ;

    sh:minCount 1 ;

    sh:maxCount 1 ;

    sh:class ex:Deck ;

    ] ;

.

ex:Bridge\_1

  a ex:Bridge ;

  smls-shacl:hasPart ex:Deck\_1 ;

  ex:currentlyServingVehicle ex:Vehicle\_1 ;

  ex:height [

      rdf:value 50.0 ;

      smls-shacl:unit unit:M ;

    ] ;

.

ex:Deck

  a rdfs:Class ;

  rdfs:subClassOf smls-shacl:PhysicalObject ;

.

ex:DeckShape

  a sh:NodeShape ;

  sh:targetClass ex:Deck ;

  sh:property [

    sh:path smls-shacl:hasPart ;

    sh:minCount 1 ;

    sh:class ex:Slab ;

    ] ;

.

ex:Deck\_1

  a ex:Deck ;

  smls-shacl:hasPart ex:Slab\_1 ;

  smls-shacl:hasPart ex:Slab\_2 ;

  smls-shacl:hasPart ex:Slab\_3 ;

.

ex:Heavy

  a ex:LoadLevelType ;

.

ex:Light

  a ex:LoadLevelType ;

.

ex:LoadLevelType

  a rdfs:Class ;

  rdfs:subClassOf smls-shacl:EnumerationType ;

.

ex:NonGeometricProperties

  a rdfs:Container ;

  rdfs:member ex:loadLevel ;

  rdfs:member ex:velocity ;

.

ex:Normal

  a ex:LoadLevelType ;

.

ex:Slab

  a rdfs:Class ;

  rdfs:subClassOf smls-shacl:PhysicalObject ;

.

ex:Slab\_1

  a ex:Slab ;

.

ex:Slab\_2

  a ex:Slab ;

.

ex:Slab\_3

  a ex:Slab ;

.

ex:Vehicle

  a rdfs:Class ;

  rdfs:subClassOf smls-shacl:PhysicalObject ;

.

ex:VehicleShape

  a sh:NodeShape ;

  sh:targetClass ex:Vehicle ;

  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:Vehicle\_1

  a ex:Vehicle ;

  ex:hasLegalSpeed true ;

  ex:loadLevel ex:Heavy ;

  ex:velocity [

      rdf:value 128.0 ;

      smls-shacl:unit unit:KiloM-PER-HR ;

      ex:accuracy [

          rdf:value 0.9 ;

        ] ;

    ] ;

.

ex:accuracy

  a rdf:Property ;

.

ex:accuracyShape

  a sh:NodeShape ;

  sh:targetObjectsOf ex:accuracy ;

  sh:class smls-shacl:QuantityValue ;

.

ex:currentlyServingVehicle

  a rdf:Property ;

.

ex:currentlyServingVehicleShape

  a sh:NodeShape ;

  sh:targetObjectsOf ex:currentlyServingVehicle ;

  sh:class smls-shacl:Vehicle ;

.

ex:hasLegalSpeed

  a rdf:Property ;

.

ex:height

  a rdf:Property ;

  smls-shacl:quantityKind quantitykind:Length ;

.

ex:heightShape

  a sh:NodeShape ;

  sh:targetObjectsOf ex:height ;

  sh:class smls-shacl:QuantityValue ;

.

ex:loadLevel

  a rdf:Property ;

.

ex:velocity

  a rdf:Property ;

  smls-shacl:quantityKind quantitykind:Speed ;

.

ex:velocityShape

  a sh:NodeShape ;

  sh:targetObjectsOf ex:velocity ;

  sh:class smls-shacl:QuantityValue ;

.

* 1. Example in SHACL (JSON-LD format)

{

  "@graph" : [ {

    "@id" : "\_:b0",

    "sh:minCount" : 1,

    "path" : "ex:height"

  }, {

    "@id" : "\_:b1",

    "class" : "ex:LoadLevelType",

    "in" : {

      "@list" : [ "ex:Light", "ex:Normal", "ex:Heavy" ]

    },

    "path" : "ex:loadLevel"

  }, {

    "@id" : "\_:b3",

    "value" : "50.0",

    "smls-shacl:unit" : {

      "@id" : "unit:M"

    }

  }, {

    "@id" : "\_:b4",

    "value" : "128.0",

    "accuracy" : "\_:b5",

    "smls-shacl:unit" : {

      "@id" : "unit:KiloM-PER-HR"

    }

  }, {

    "@id" : "\_:b5",

    "value" : "0.9"

  }, {

    "@id" : "\_:b6",

    "datatype" : "xsd:boolean",

    "path" : "ex:hasLegalSpeed"

  }, {

    "@id" : "\_:b7",

    "class" : "ex:Slab",

    "sh:minCount" : 1,

    "path" : "smls-shacl:hasPart"

  }, {

    "@id" : "\_:b8",

    "class" : "ex:Deck",

    "sh:maxCount" : 1,

    "sh:minCount" : 1,

    "path" : "smls-shacl:hasPart"

  }, {

    "@id" : "<https://w3id.org/def/smls-example-shacl>",

    "@type" : "owl:Ontology",

    "imports" : "<https://w3id.org/def/smls-shacl>"

  }, {

    "@id" : "ex:Bridge",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-shacl:PhysicalObject",

    "definition" : {

      "@language" : "nl",

      "@value" : "Verbinding voor verkeer tussen twee landhoofden die gescheiden wordt door water."

    },

    "prefLabel" : [ {

      "@language" : "nl",

      "@value" : "Brug"

    }, {

      "@language" : "en",

      "@value" : "Bridge"

    } ]

  }, {

    "@id" : "ex:BridgeShape",

    "@type" : "sh:NodeShape",

    "property" : [ "\_:b8", "\_:b0" ],

    "targetClass" : "ex:Bridge"

  }, {

    "@id" : "ex:Bridge\_1",

    "rdf:type" : {

      "@id" : "ex:Bridge"

    },

    "currentlyServingVehicle" : "ex:Vehicle\_1",

    "height" : "\_:b3",

    "hasPart" : "ex:Deck\_1"

  }, {

    "@id" : "ex:Deck",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-shacl:PhysicalObject"

  }, {

    "@id" : "ex:DeckShape",

    "@type" : "sh:NodeShape",

    "property" : "\_:b7",

    "targetClass" : "ex:Deck"

  }, {

    "@id" : "ex:Deck\_1",

    "rdf:type" : {

      "@id" : "ex:Deck"

    },

    "hasPart" : [ "ex:Slab\_3", "ex:Slab\_2", "ex:Slab\_1" ]

  }, {

    "@id" : "ex:Heavy",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:Light",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:LoadLevelType",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-shacl:EnumerationType"

  }, {

    "@id" : "ex:NonGeometricProperties",

    "@type" : "rdfs:Container",

    "member" : [ "ex:velocity", "ex:loadLevel" ]

  }, {

    "@id" : "ex:Normal",

    "rdf:type" : {

      "@id" : "ex:LoadLevelType"

    }

  }, {

    "@id" : "ex:Slab",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-shacl:PhysicalObject"

  }, {

    "@id" : "ex:Slab\_1",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Slab\_2",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Slab\_3",

    "rdf:type" : {

      "@id" : "ex:Slab"

    }

  }, {

    "@id" : "ex:Vehicle",

    "@type" : "rdfs:Class",

    "subClassOf" : "smls-shacl:PhysicalObject"

  }, {

    "@id" : "ex:VehicleShape",

    "@type" : "sh:NodeShape",

    "property" : [ "\_:b6", "\_:b1" ],

    "targetClass" : "ex:Vehicle"

  }, {

    "@id" : "ex:Vehicle\_1",

    "rdf:type" : {

      "@id" : "ex:Vehicle"

    },

    "ex:hasLegalSpeed" : true,

    "loadLevel" : "ex:Heavy",

    "velocity" : "\_:b4"

  }, {

    "@id" : "ex:accuracy",

    "@type" : "rdf:Property"

  }, {

    "@id" : "accuracy:Shape",

    "@type" : "sh:NodeShape",

    "class" : "smls-shacl:QuantityValue",

    "targetObjectsOf" : "ex:accuracy"

  }, {

    "@id" : "ex:currentlyServingVehicle",

    "@type" : "rdf:Property"

  }, {

    "@id" : "currentlyServingVehicle:Shape",

    "@type" : "sh:NodeShape",

    "class" : "smls-shacl:Vehicle",

    "targetObjectsOf" : "ex:currentlyServingVehicle"

  }, {

    "@id" : "ex:hasLegalSpeed",

    "@type" : "rdf:Property"

  }, {

    "@id" : "ex:height",

    "@type" : "rdf:Property",

    "quantityKind" : "quantitykind:Length"

  }, {

    "@id" : "height:Shape",

    "@type" : "sh:NodeShape",

    "class" : "smls-shacl:QuantityValue",

    "targetObjectsOf" : "ex:height"

  }, {

    "@id" : "ex:loadLevel",

    "@type" : "rdf:Property"

  }, {

    "@id" : "ex:velocity",

    "@type" : "rdf:Property",

    "quantityKind" : "quantitykind:Speed"

  }, {

    "@id" : "velocity:Shape",

    "@type" : "sh:NodeShape",

    "class" : "smls-shacl:QuantityValue",

    "targetObjectsOf" : "ex:velocity"

  } ],

  "@context" : {

    "minCount" : {

      "@id" : "<http://www.w3.org/ns/shacl#minCount>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#integer>"

    },

    "path" : {

      "@id" : "<http://www.w3.org/ns/shacl#path>",

      "@type" : "@id"

    },

    "in" : {

      "@id" : "<http://www.w3.org/ns/shacl#in>",

      "@type" : "@id"

    },

    "class" : {

      "@id" : "<http://www.w3.org/ns/shacl#class>",

      "@type" : "@id"

    },

    "unit" : "<http://qudt.org/vocab/unit/>",

    "value" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#value>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#decimal>"

    },

    "targetObjectsOf" : {

      "@id" : "<http://www.w3.org/ns/shacl#targetObjectsOf>",

      "@type" : "@id"

    },

    "accuracy" : {

      "@id" : "<https://w3id.org/def/smls-example-shacl#accuracy>",

      "@type" : "@id"

    },

    "datatype" : {

      "@id" : "<http://www.w3.org/ns/shacl#datatype>",

      "@type" : "@id"

    },

    "quantityKind" : {

      "@id" : "<https://w3id.org/def/smls-shacl#quantityKind>",

      "@type" : "@id"

    },

    "subClassOf" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#subClassOf>",

      "@type" : "@id"

    },

    "member" : {

      "@id" : "<http://www.w3.org/2000/01/rdf-schema#member>",

      "@type" : "@id"

    },

    "property" : {

      "@id" : "<http://www.w3.org/ns/shacl#property>",

      "@type" : "@id"

    },

    "targetClass" : {

      "@id" : "<http://www.w3.org/ns/shacl#targetClass>",

      "@type" : "@id"

    },

    "imports" : {

      "@id" : "<http://www.w3.org/2002/07/owl#imports>",

      "@type" : "@id"

    },

    "maxCount" : {

      "@id" : "<http://www.w3.org/ns/shacl#maxCount>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#integer>"

    },

    "definition" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#definition>"

    },

    "prefLabel" : {

      "@id" : "<http://www.w3.org/2004/02/skos/core#prefLabel>"

    },

    "height" : {

      "@id" : "<https://w3id.org/def/smls-example-shacl#height>",

      "@type" : "@id"

    },

    "currentlyServingVehicle" : {

      "@id" : "<https://w3id.org/def/smls-example-shacl#currentlyServingVehicle>",

      "@type" : "@id"

    },

    "hasPart" : {

      "@id" : "<https://w3id.org/def/smls-shacl#hasPart>",

      "@type" : "@id"

    },

    "velocity" : {

      "@id" : "<https://w3id.org/def/smls-example-shacl#velocity>",

      "@type" : "@id"

    },

    "loadLevel" : {

      "@id" : "<https://w3id.org/def/smls-example-shacl#loadLevel>",

      "@type" : "@id"

    },

    "hasLegalSpeed" : {

      "@id" : "<https://w3id.org/def/smls-example-shacl#hasLegalSpeed>",

      "@type" : "<http://www.w3.org/2001/XMLSchema#boolean>"

    },

    "rest" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>",

      "@type" : "@id"

    },

    "first" : {

      "@id" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns#first>",

      "@type" : "@id"

    },

    "quantitykind" : "<http://qudt.org/vocab/quantitykind/>",

    "qudt" : "<http://qudt.org/schema/qudt/>",

    "ex" : "<https://w3id.org/def/smls-example-shacl>#",

    "smls-shacl" : "<https://w3id.org/def/smls-shacl>#",

    "owl" : "<http://www.w3.org/2002/07/owl>#",

    "rdf" : "<http://www.w3.org/1999/02/22-rdf-syntax-ns>#",

    "sh" : "<http://www.w3.org/ns/shacl>#",

    "xsd" : "<http://www.w3.org/2001/XMLSchema>#",

    "skos" : "<http://www.w3.org/2004/02/skos/core>#",

    "rdfs" : "<http://www.w3.org/2000/01/rdf-schema>#",

    "dash" : "<http://datashapes.org/dash>#"

  }

}

1. (informative)  
     
   Relationships with other asset/product modelling standards
   1. General

This annex describes the relationships between the SML standard and two directly related standard: ISO 21597 (ICDD) and ISO 23387 (DT).

* 1. Relationship with ISO 21597

ISO 21597-1, Information Container for Document Delivery (ICDD), defines a standard ‘linked data’-based way to package interrelated documents for mostly human interpretation.

It specifies a “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 data exchange (here referred to as “delivery”), not on data sharing. However, both the container description and the document and link descriptions could also be reused in the context of data 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 ín 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 data sets and data models).

In the near future there will also be an ICDD Part 2, ISO 21597-2, that will offer specializations for the generic document links from Part 1.

SMLS 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 SMLS;
* The ICDD Part 1 ct:Document class is similar to the SMLS Top Level “InformationObject” concept.

NOTE 1 Although ICDD assumes CWA it uses primarily OWA-based OWL.

NOTE 2 The objectified modelling of relations in ICDD makes it hard to reuse existing linked data functionalities like for transitivity, symmetry, inverses and functional/inverse-functional characteristics.

* 1. Relationship with ISO 23387
     1. Introduction

According to ISO 23387, data templates (DTs) are instances of the 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 DTs are dealt with via references from IFC (SPFF or XML) files according to ISO 16739-1.

According to this document, data templates (DTs) 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 data instance files.

Key advantages using the approach in this document over the 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 APIs avoiding software vendor lock-in;
* Increased modelling power for advanced data modelling involving i.e. complex constraints and derivations;
* Standard mechanisms for linking on data set level, data model level and between (elements of) data sets and data models;
* IFC-independent asset/product data instantiation.
  + 1. Modelling relationship with ISO 12006-3

Figure G.1 shows that ISO 12006-3 is *conceptually* fully complementary to the scope of this document.

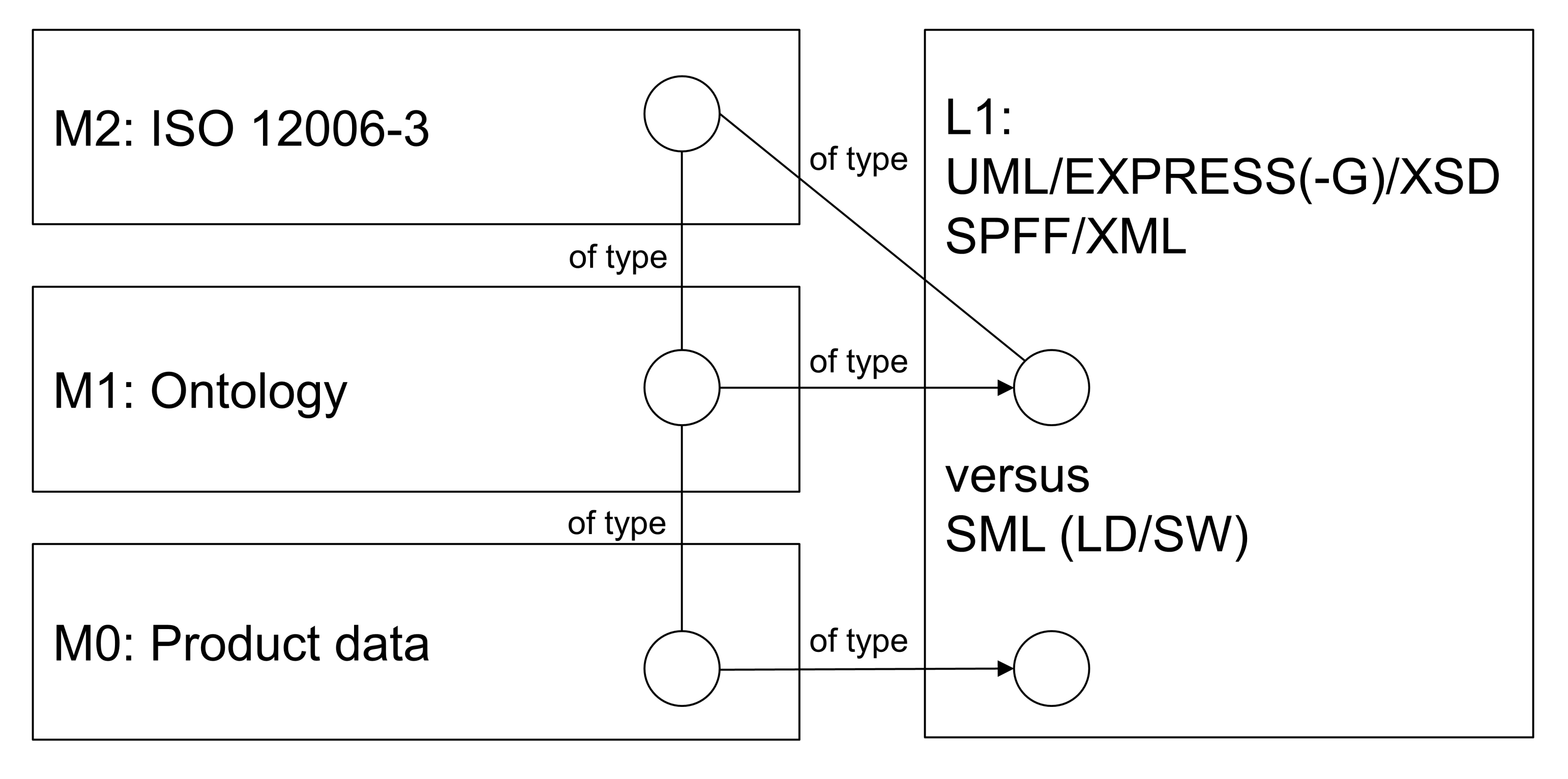


Figure G.1 — ISO 12006-3 versus SML

The ISO 12006-3 schema is a M2: knowledge model on top of a M1: Ontology. Formulated the other way round, the instances of ISO 12006-3 are ontology elements.

The main difference is in the underlying L1 data language (ISO 12006-3 using OMG UML /ISO STEP/W3C XML technology, this document using W3C LD/SW technology).

* + 1. Example CRB

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

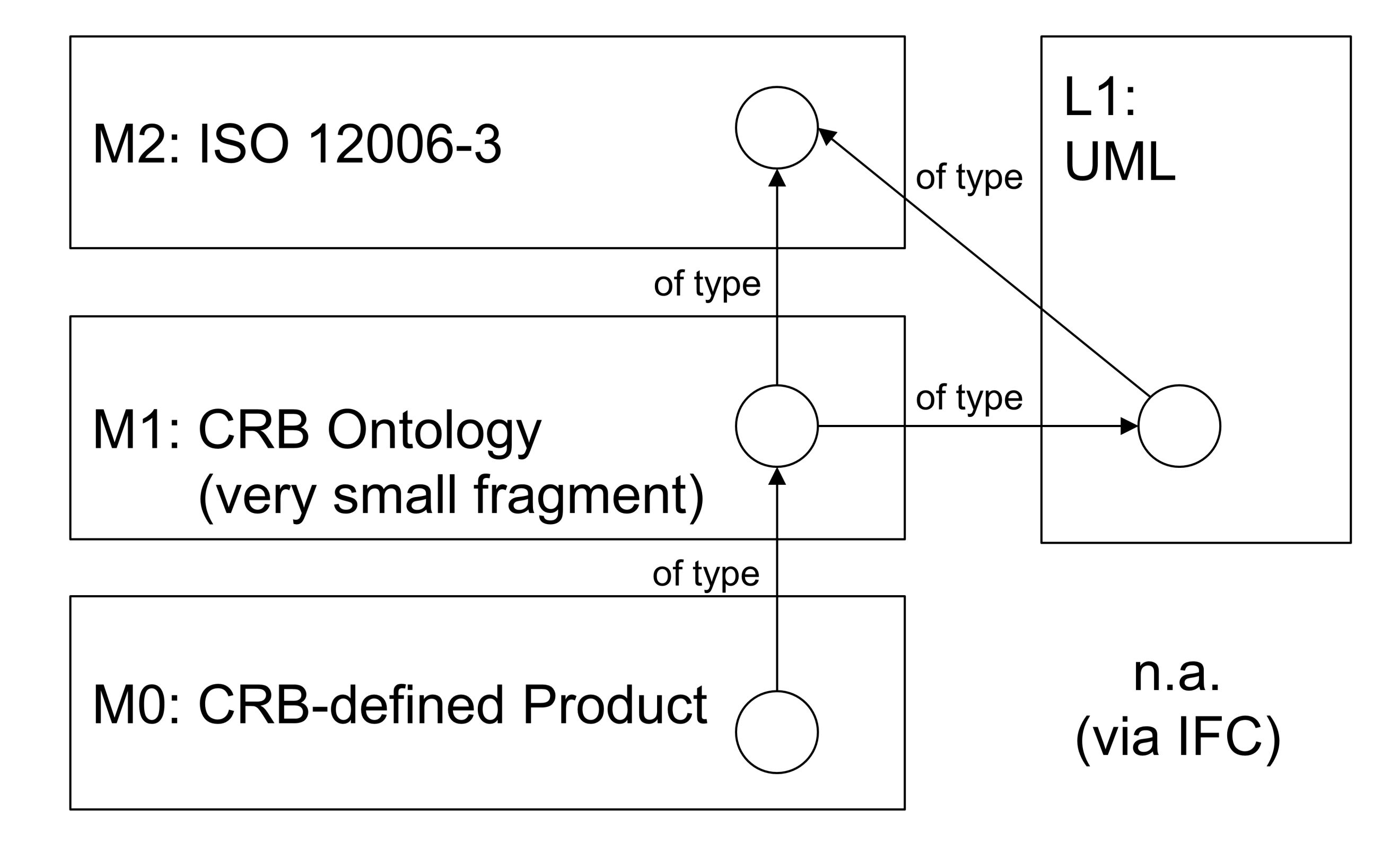


Figure G.2 — CRB example according to ISO 12006-3

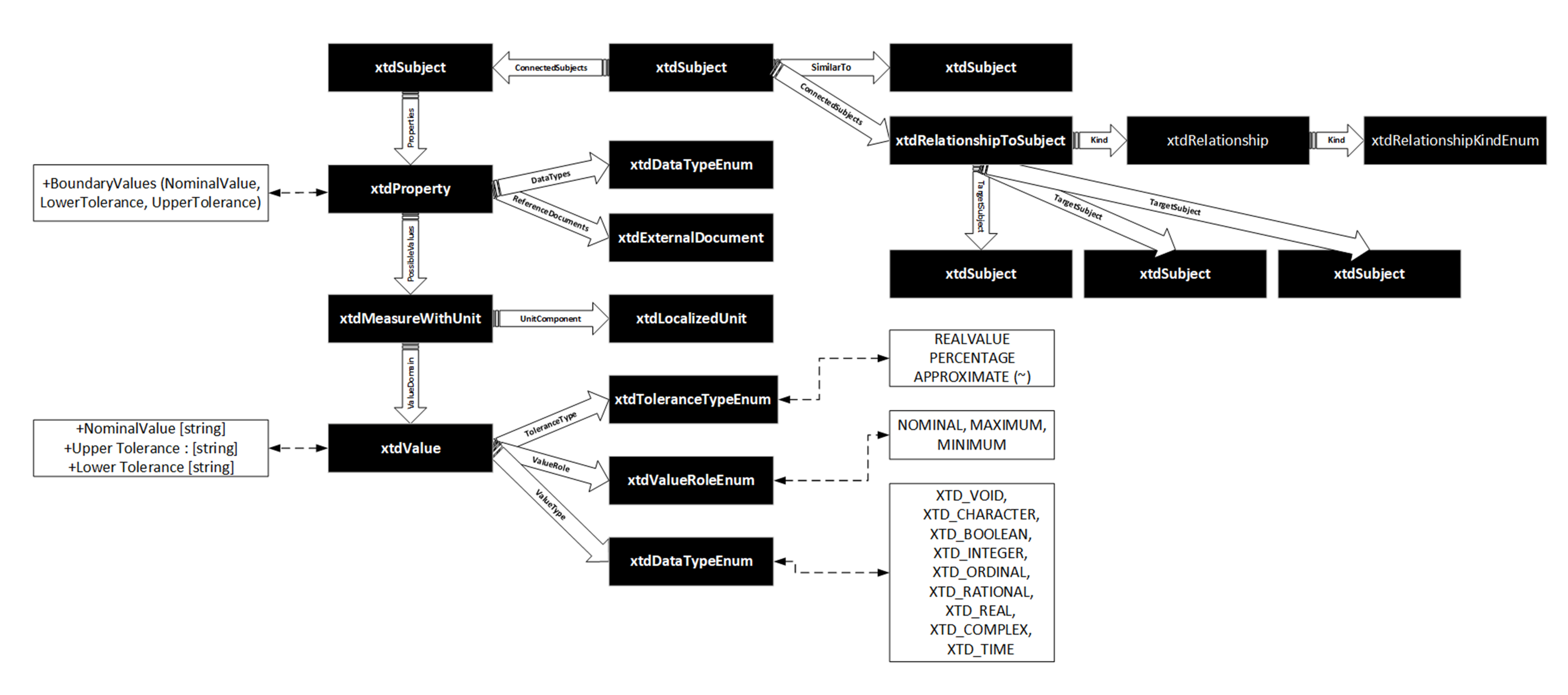


Figure G.3 —Relevant ISO 12006-3 fragment in UML-like format

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 G.4). This quantity is measured in “kN/m2” 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 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 value “0.8”;
* ValueRole gets value “MINIMUM”;
* ValueType gets value “XTD\_REAL”;
* Order gets value “0”.

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

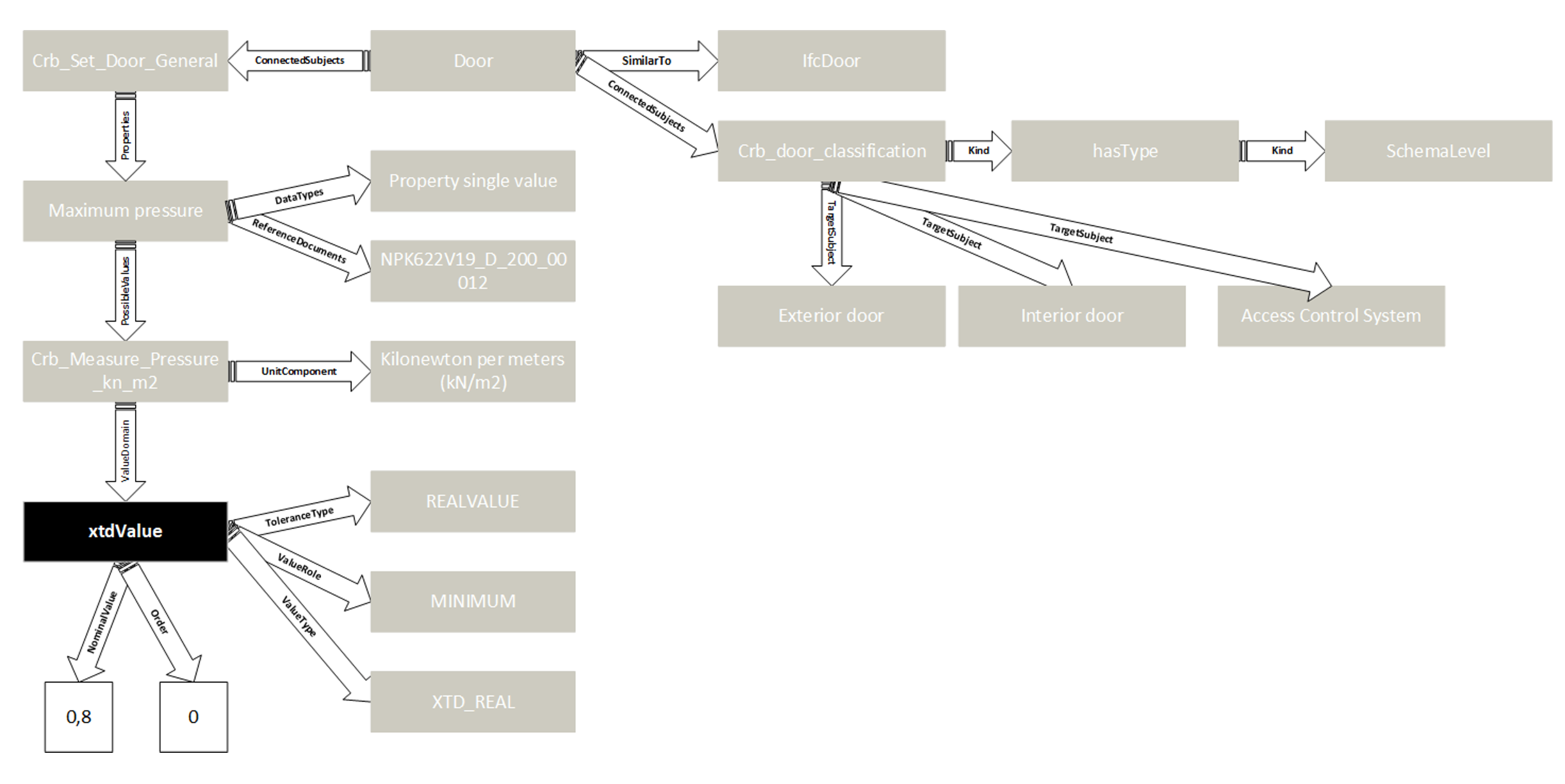


Figure G.4 — CRB ontology fragment (schematically)

The example is now extended in two ways (figure G.5):

* A boolean attribute “isExternal” is added including a constraint for exterior door (being true);
* 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 G.5 — Extending the example

For computer-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: data language to e.g. be able to explicitly model the relevant minimum value in a closed world fashion so that product instantiations can be automatically verified. As data format Turtle is used. This situation is pictured in figure G.6.

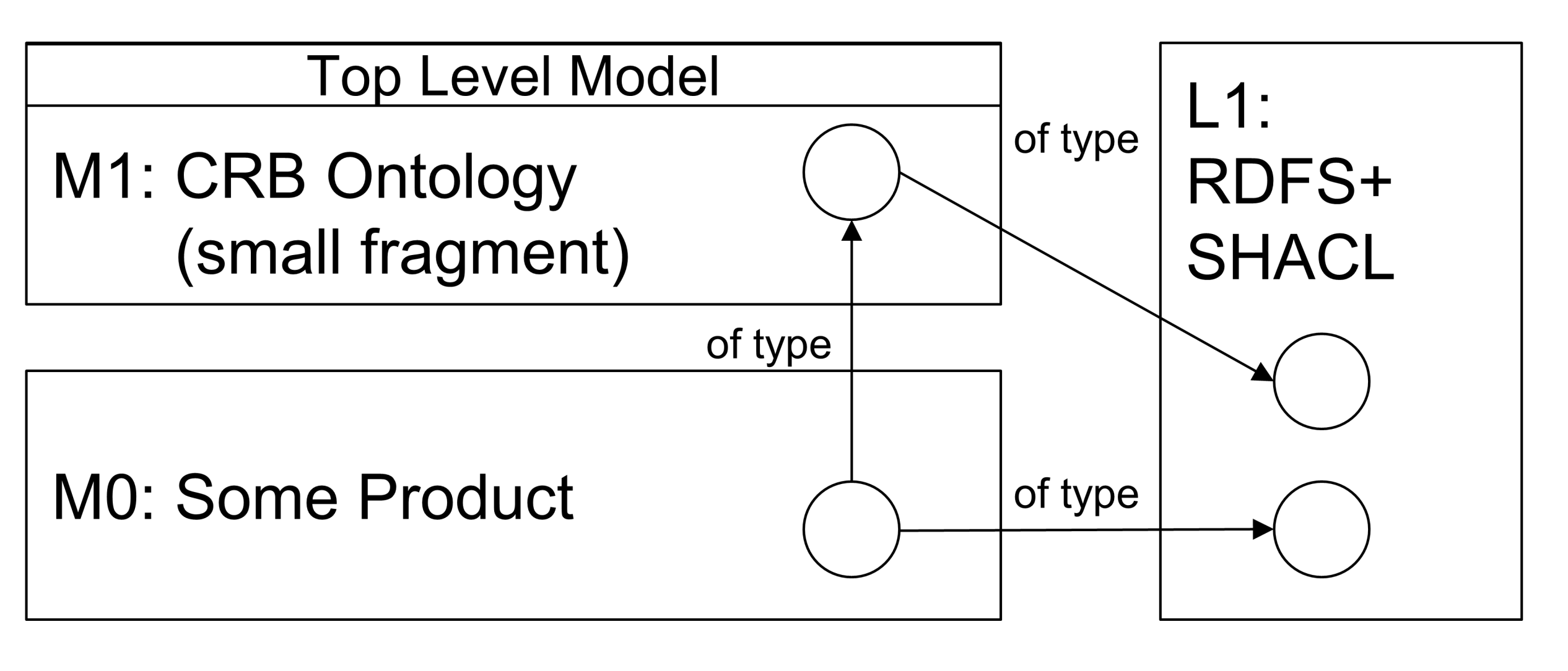


Figure G.6 — Mapped to SML

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

# baseURI: https://w3id.org/def/crb

# imports: https://w3id.org/def/smls-shacl

# prefix: crb

@prefix crb: <https://w3id.org/def/crb#> .

@prefix dash: <http://datashapes.org/dash#> .

@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 smls-shacl: <https://w3id.org/def/smls-shacl#> .

@prefix unit: <http://qudt.org/vocab/unit/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/def/crb>

a owl:Ontology ;

owl:imports <https://w3id.org/def/smls-shacl> ;

.

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

crb:Door

a rdfs:Class ;

rdfs:subClassOf smls-shacl: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 ;

.

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 as (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 ;

smls-shacl:unit unit:KiloPA ;

rdfs:comment "D.200.00012 (NPK 622, 2019)" ;

rdfs:seeAlso <https://npkviewer.crb.ch> ;

.

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

Next, the boolean isExternal is defined:

crb:isExternal

a rdf:Property ;

.

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

crb:eBKP-H

a rdf:Property ;

rdfs:range crb:EBKP-HType ;

.

crb:EBKP-HType

a rdfs:Class ;

rdfs:subClassOf smls-shacl:EnumerationType ;

.

crb:D32

a crb:EBKP-HType ;

.

crb:E32

a crb:EBKP-HType ;

.

crb:G14

a crb:EBKP-HType ;

.

NOTE 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).

Next we define an attribute of doors that is a standard quantity analysis for door construction costs evaluations (also not in UML). 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 ;

smls-shacl:abbreviation "FII"@ch-de ;

smls-shacl:abbreviation "SPINT"@ch-fr ;

smls-shacl:unit unit:M2 ;

rdfs:comment "eBKP-H Anwenderhandbuch, 2012" ;

rdfs:seeAlso <https://www.crb.ch/Normen-Standards/Baukostenplaene/eBKP\_H.html> ;

.

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 ;

smls-shacl:abbreviation "FII"@ch-de ;

smls-shacl:abbreviation "SPINT"@ch-fr ;

smls-shacl:unit unit:M2 ;

rdfs:comment "eBKP-H Anwenderhandbuch, 2012" ;

rdfs:seeAlso <https://www.crb.ch/Normen-Standards/Baukostenplaene/eBKP\_H.html> ;

.

The “access control system” (ACS) is defined by:

crb:AccessControlSystem

a rdfs:Class ;

rdfs:subClassOf smls-shacl: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 controle d`accés"@ch-fr ;

skos:prefLabel "Zutrittskontrollanlage"@ch-de ;

smls-shacl: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 raccordemen"@ch-fr ;

smls-shacl:abbreviation "AAZK"@ch-de ;

smls-shacl: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-H ;

sh:class crb:EBKP-HType ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:property [

rdf:type sh:PropertyShape ;

sh:path crb:isControlledBy ;

sh:class crb:AccessControlSystem ;

sh:maxCount 1 ;

sh:minCount 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:decimal ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:property [

a sh:PropertyShape ;

sh:path crb:eBKP-H ;

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:decimal ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:property [

a sh:PropertyShape ;

sh:path crb:eBKP-H ;

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

) ;

sh:datatype xsd:decimal ;

sh:minInclusive 0.8 ;

] ;

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\_H ;

sh:class crb:EBKP\_HType ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:property [

a sh:PropertyShape ;

sh:path crb:accessControlSystemConnectionPointCount ;

sh:datatype xsd:integer ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:property [

a sh:PropertyShape ;

sh:path crb:eBKP-H ;

sh:class crb:EBKP-HType ;

sh:hasValue crb:D32 ;

] ;

sh:targetClass crb:AccessControlSystem ;

.

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

crb:isControlledBy

a rdf:Property ;

.

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

crb:Door\_1

rdf:type crb:AcceptableMechanicalPressureLoadDoor ;

crb:eBKP-H crb:E32 ;

crb:exteriorDoorSurfaceArea [

rdf:value 2.4 ;

] ;

crb:isControlledBy crb:ACS\_1 ;

crb:isExternal true ;

crb:maxPressure [

rdf:value 0.5 ;

] ;

.

crb:ACS\_1

a crb:AccessControlSystem ;

crb:accessControlSystemConnectionPointCount 5 ;

crb:eBKP-H crb:D32 ;

.

This data can be automatically verified given these specific instances (Figure G.7).

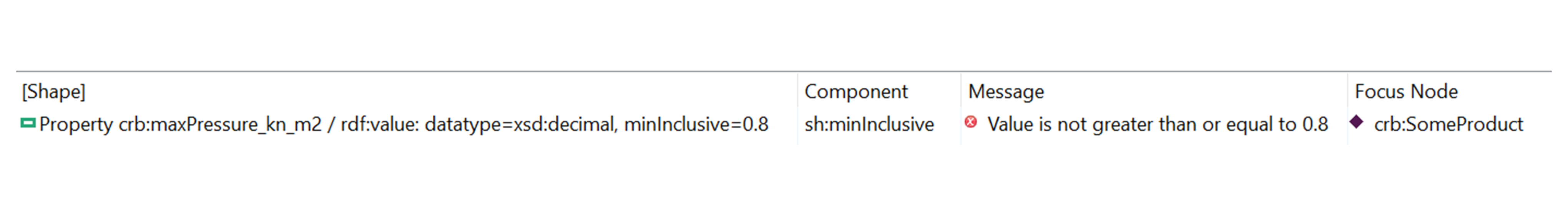


Figure G.7 — ISO 12006-3 versus SML

In the future, when ISO 12006-3 is also available in a linked data format (say SHACL) two approaches (ISO 12006-3 and SMLS) could be combined as depicted in Figure G.8.

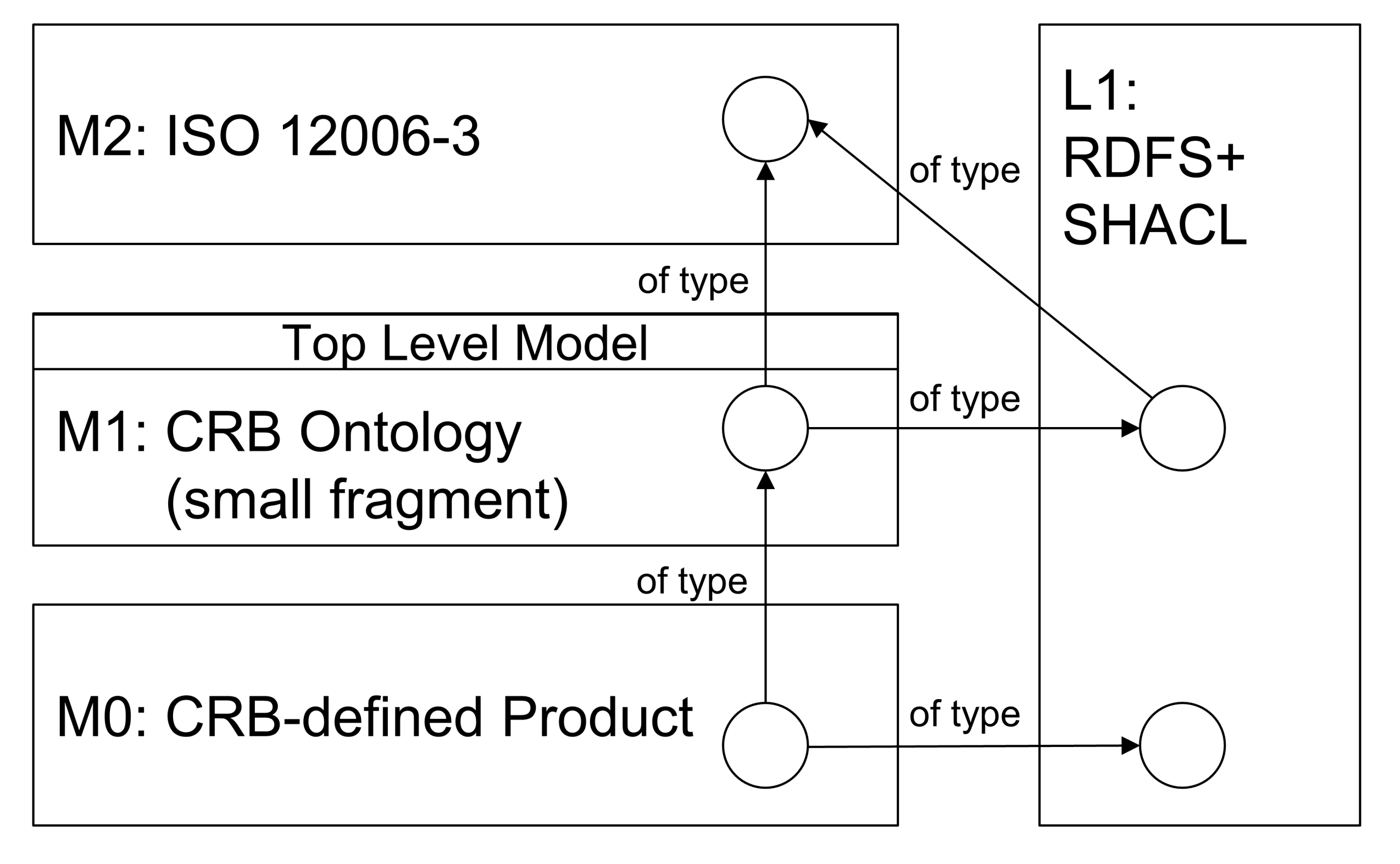


Figure G.8 — Future combination potential

This way, elements in the CRB Ontology will be double typed: towards their ISO 12006-3 model types 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 ;

…

.

* + 1. Full CRB code example in SHACL (in Turtle format)

# baseURI: https://w3id.org/def/crb

# imports: https://w3id.org/def/smls-shacl

# prefix: crb

@prefix crb: <https://w3id.org/def/crb#> .

@prefix dash: <http://datashapes.org/dash#> .

@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 smls-shacl: <https://w3id.org/def/smls-shacl#> .

@prefix unit: <http://qudt.org/vocab/unit/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/def/crb>

a owl:Ontology ;

owl:imports <https://w3id.org/def/smls-shacl> ;

.

crb:ACS\_1

a crb:AccessControlSystem ;

crb:accessControlSystemConnectionPointCount 5 ;

crb:eBKP-H 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:decimal ;

sh:minInclusive 0.8 ;

] ;

sh:targetClass crb:AcceptableMechanicalPressureLoadDoor ;

.

crb:AccessControlSystem

a rdfs:Class ;

rdfs:subClassOf smls-shacl: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 controle d`accés"@ch-fr ;

skos:prefLabel "Zutrittskontrollanlage"@ch-de ;

smls-shacl:abbreviation "ACS"@en ;

.

crb:AccessControlSystemShape

a sh:NodeShape ;

sh:property [

a sh:PropertyShape ;

sh:path crb:accessControlSystemConnectionPointCount ;

sh:datatype xsd:integer ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:property [

a sh:PropertyShape ;

sh:path crb:eBKP-H ;

sh:class crb:EBKP-HType ;

sh:hasValue crb:D32 ;

] ;

sh:property [

a sh:PropertyShape ;

sh:path crb:eBKP-H ;

sh:class crb:EBKP-HType ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:targetClass crb:AccessControlSystem ;

.

crb:D32

a crb:EBKP-HType ;

.

crb:Door

a rdfs:Class ;

rdfs:subClassOf smls-shacl: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-H ;

sh:class crb:EBKP-HType ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:property [

a sh:PropertyShape ;

sh:path crb:isControlledBy ;

sh:class crb:AccessControlSystem ;

sh:maxCount 1 ;

sh:minCount 1 ;

] ;

sh:targetClass crb:Door ;

.

crb:Door\_1

a crb:AcceptableMechanicalPressureLoadDoor ;

crb:eBKP-H crb:E32 ;

crb:exteriorDoorSurfaceArea [

rdf:value 2.4 ;

] ;

crb:isControlledBy crb:ACS\_1 ;

crb:isExternal true ;

crb:maxPressure [

rdf:value 0.5 ;

] ;

.

crb:E32

a crb:EBKP-HType ;

.

crb:EBKP-HType

a rdfs:Class ;

rdfs:subClassOf smls-shacl: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-H ;

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:decimal ;

sh:maxCount 1 ;

sh:minCount 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-H ;

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:decimal ;

sh:maxCount 1 ;

sh:minCount 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 raccordemen"@ch-fr ;

smls-shacl:abbreviation "AAZK"@ch-de ;

smls-shacl:abbreviation "NPRIC"@ch-fr ;

.

crb:eBKP-H

a rdf:Property ;

.

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 ;

smls-shacl:abbreviation "FTT"@ch-de ;

smls-shacl:abbreviation "SPOR"@ch-fr ;

smls-shacl:unit unit:M2 ;

.

crb:interiorDoorSurfaceArea

a rdf:Property ;

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 ;

smls-shacl:abbreviation "FII"@ch-de ;

smls-shacl:abbreviation "SPINT"@ch-fr ;

smls-shacl:unit unit:M2 ;

.

crb:isControlledBy

a rdf:Property ;

.

crb:isExternal

a rdf:Property ;

.

crb:maxPressure

a rdf:Property ;

rdfs:comment "D.200.00012 (NPK 622, 2019)" ;

rdfs:seeAlso <https://npkviewer.crb.ch> ;

smls-shacl:unit unit:KiloPA ;

.

Bibliography

[1] W3C Linked Data/Semantic Web (LD/SW)  
<https://www.w3.org/standards/semanticweb/data>

[2] QUDT by [qudt.org](file:///C:/Users/eal/AppData/Local/Temp/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)](https://w3c.github.io/shacl/shacl-af/)

[6] Defining N-ary Relations on the Semantic Web

<https://www.w3.org/TR/swbp-n-aryRelations/>

[7] 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*

[8] ISO 19650-2, *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling - Part 2: Delivery phase of the assets*

[9] ISO 19650-3, *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling - Part 3: Operational phase of assets*

[10] ISO 19650-4, *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*

[11] ISO 19650-5, *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling - Part 5: Security-minded approach to information management*

[12] ISO 11354-1, *Advanced automation technologies and their applications — Requirements for establishing manufacturing enterprise process interoperability — Part 1: Framework for enterprise interoperability*

[13] ISO/IEC 80000, *Quantities and units*

1. ) A common principle by go-fair.org in today’s data management. [↑](#footnote-ref-2)
2. ) The 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(en), 3.1.65]. [↑](#footnote-ref-3)
3. ) From now referred to as just “OWL”. [↑](#footnote-ref-4)
4. ) When there is a domain *and* a range: mapping to a NodeShape with a PropertyShape having a sh:targetClass. When there is a range only: mapping to a NodeShape with sh:targetObjectsFrom. [↑](#footnote-ref-5)