Date: YYYY-MM-DD

Industrial Data Ontology

WD stage

Warning for WDs and CDs

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO [had/had not] received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee [or Project Committee] ISO/TC [or ISO/PC] ###, [name of committee], Subcommittee SC ##, [name of subcommittee].

This second/third/… edition cancels and replaces the first/second/… edition (ISO #####:####), which has been technically revised.

The main changes are as follows:

— xxx xxxxxxx xxx xxxx

A list of all parts in the ISO ##### series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

POSC Caesar Association
Evolve IT Fornebu
Martin Lingesvei 25
1364 FORNEBU
NORWAY

General Manager Nils Sandsmark
Mobile: +47-95 85 32 20
E-mail: nils.sandsmark@posccaesar.org
Phone: +41 22 749 01 11
Website: https://www.posccaesar.org/

Contents

Figures

Introduction

An engineering asset, such as a new offshore renewables energy field, an existing sub-sea development, or an aircraft model, is a system of systems. Each system has multiple stakeholders, and each stakeholder has specified the use of structured vocabularies (e.g., from ISO, IEC, ASME, DIN), data model(s) and design standards. The exchange of information at each stage of the life cycle, and within a life cycle phase, such as design, is therefore complicated. Errors in information exchange can lead to risks, such as cost overruns, safety events and product delays. Management of these risks include controls such as facilitating automated data exchange. This semantic interoperability involves computers being able to unambiguously understand data and its context and is achieved through the reasoning ability of an ontology.

IDO is an OWL 2 ontology (cf. 3.20). IDO is designed to support machine automated reasoning over information used in the design and through-life operation of complex, long-life, engineering assets.

IDO is suitable for industrial use cases, to create vocabularies and asset models, and exploit OWL DL reasoning (cf. 3.21) for quality assurance and inference of implicit knowledge.

IDO provides "usable" guidance for creating ontology patterns and examples of patterns commonly used to capture important data and relations in the engineering design process. The use, and reuse, of approved patterns reduces risks that modelling mistakes will be made and increases the speed of the model development and quality checking processes.

The intended users of IDO are involved in the design, construction, manufacturing, operation, maintenance, and disposal of assets and discrete or continuous processing systems in the engineering sector.

IDO was developed in response to demand from the industrial engineering community for ontology-based solutions to the problem of semantic interoperability across information systems associated with the multiple stakeholders involved in complex systems engineering projects and through life asset management.

IDO has a track-record of use in the oil and gas sector under the title of ISO TR 15926-14. It has been used for OWL DL reasoning over material master data in the design phase as well as for representing as-built process plants with millions of parts. It enables annotation of work orders for use in language models. IDO is also coherent with, and supports, modelling done with IEC 81346-1:2022 Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 1: Basic rules and its sub-parts which provides an important link for the users of these standards to the models on which the data they are referring to in IEC 81346 is based.

IDO draws on the heritage of existing ontologies published (or in the process of being published) by ISO/IEC such as ISO/IEC 21838-1:2021) Information technology – Top-level ontologies (TLO) – Part 1: Requirements; ISO/IEC 21838-2:2021 (Top-level ontologies (TLO) – Part 2: Basic Formal Ontology (BFO)), ISO/IEC DIS 21838-3 – Top-level ontologies (TLO) – Part 3: Descriptive ontology for linguistic and cognitive engineering (DOLCE)), ISO 15926-2:2003 (reconfirmed 2018) Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 2: Data model; and ISO/TS 15926-12:2018 Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 12: Life-cycle integration ontology represented in Web Ontology Language (OWL).

What IDO provides its industrial stakeholder community is a "usable" ontology. IDO provides 1) language that is accessible to engineers and 2) industrially relevant modelling patterns, while retaining the technical capability to perform OWL DL reasoning with a well-founded ontological base. This combination of user-centric characteristics is the primary motivation for the standardisation of IDO.

Figure 1 provides an overview of the classes in IDO (on reading diagrams, cf. section 5.1). Figure 2 provides an example of a pattern in this case for describing the purpose of a typical engineering asset (a compressor) and the associated classes and relations necessary to distinguish it from other engineering assets.

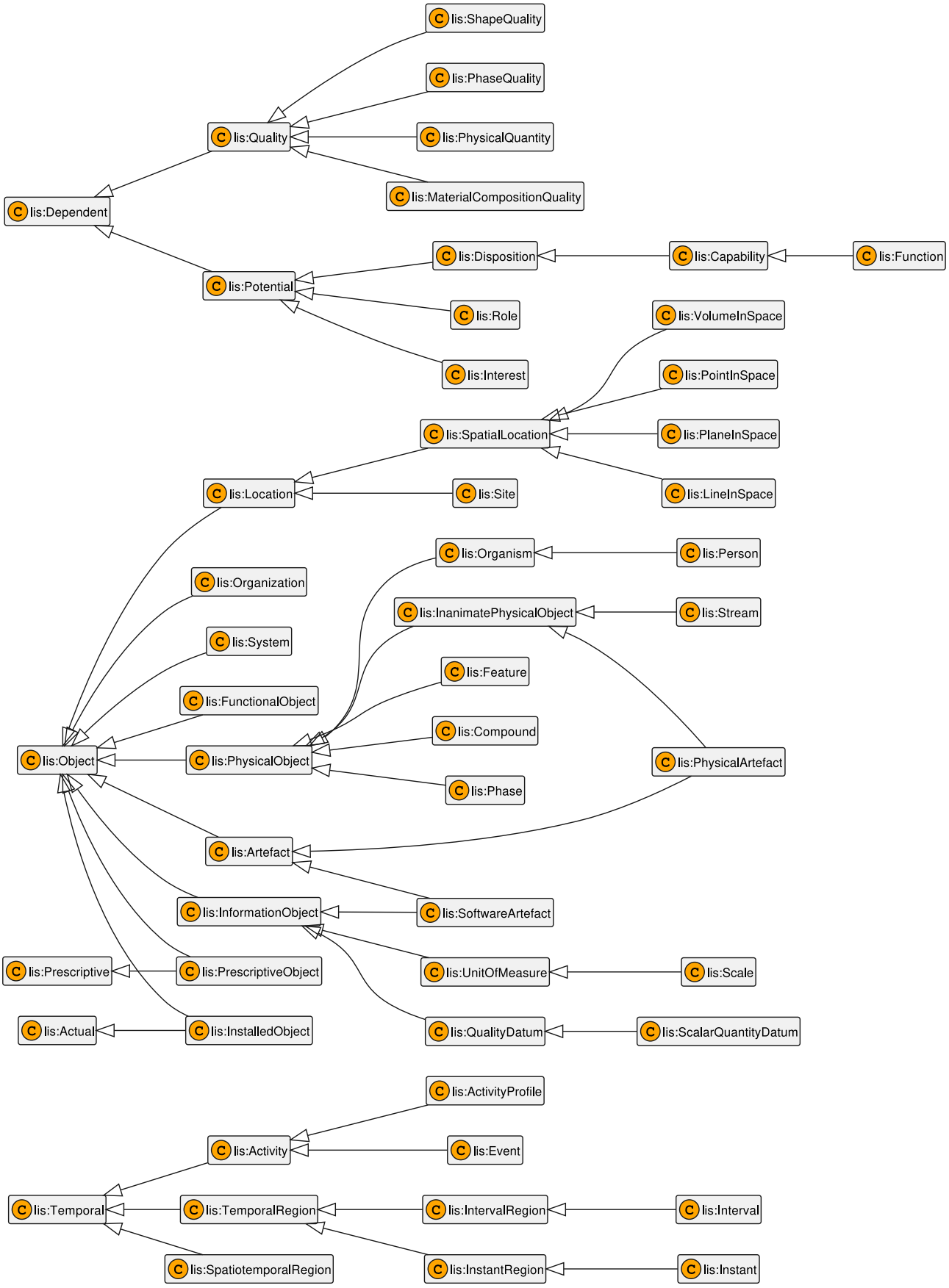


Figure 1 — The full IDO class hierarchy

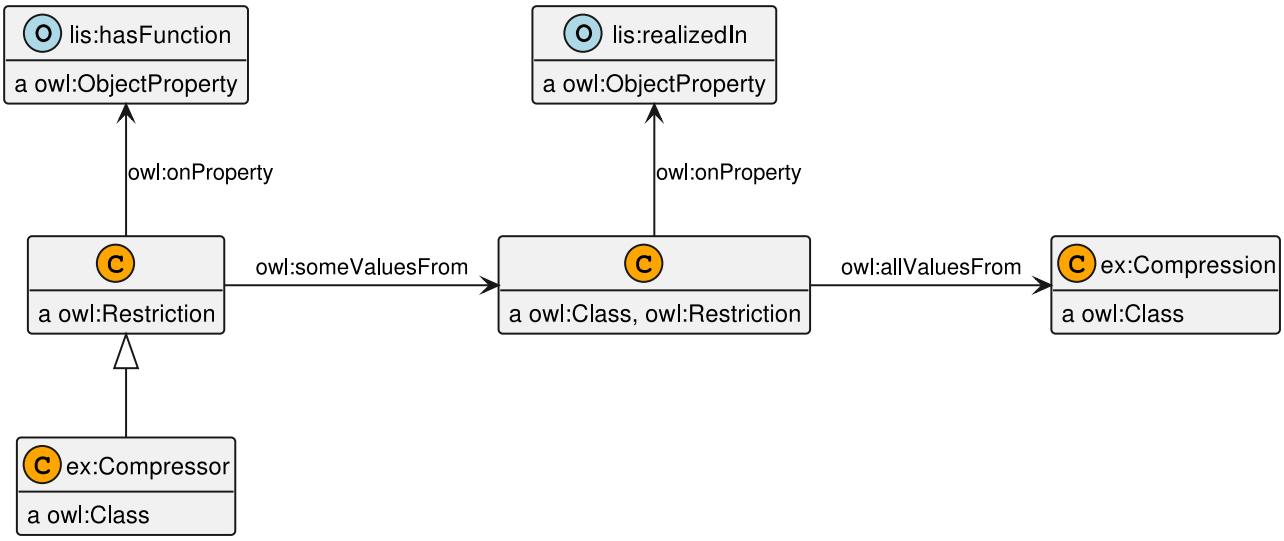


Figure 2 — IDO pattern for the function of a class of artefacts (in this case, *Compressor*)

1 Scope

This document describes the Industrial Data Ontology (IDO), a Web Ontology Language (OWL) ontology.

IDO is intended for industrial data and information, to build vocabularies and manage asset models which employ reference data libraries and exploit OWL DL. IDO is intended for use in all life cycle phases of industrial assets and processes.

The following are within the scope of this document:

- general overview of IDO and of the main principles
- normative definitions of IDO classes and relations
- informative examples of the use of IDO terminology in context
- re-usable representation patterns

The following are outside the scope of this document:

- specification of data modelling languages, such as RDF and OWL used in ontology development
- specification of methods for reasoning with ontologies

2 Normative references

The normative references shall be introduced by the following wording.

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

ISO ##### #, General title — Part #: Title of part

ISO ##### ##:20##, General title — Part ##: Title of part

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 21838-1:2021 Information technology — Top-level ontologies (TLO) — Part 1: Requirements and the following apply.

ISO and IEC maintain terminology databases for use in standardisation at the following addresses:

ISO Online browsing platform: available at https://www.iso.org/obp

IEC Electropedia: available at https://www.electropedia.org/

3.1
entity
item that is perceivable or conceivable

Note 1 to entry: The term 'entity' is a catch-all terms analogous to 'something'. In ontology circles, 'entity' and 'thing' are commonly used.

[SOURCE: ISO 1087-1:2000] and [SOURCE: ISO/IEC 21838-1] modified as per below.

3.2
class
general entity

Note 1 to entry: In some ontology communities, all general entities are referred to as classes. In other ontology communities, a distinction is drawn between classes as the extensions of general entities (for example as sets of instances) and the general entities themselves, sometimes referred to as 'types', 'kinds', or 'universals'. The expression 'class or type' is used in what follows in order to remain neutral as between these different usages.

[SOURCE: ISO/IEC 21838-1]

3.3
individual
individual entity

Note 1 to entry: In contrast to classes or types, individuals are not exemplified or instantiated by further entities.

Extract from [SOURCE: ISO/IEC 21838-1]

3.4
relation
way in which entities are related

Note 1 to entry: Relations can hold between individuals (this motor is part of this pump system); or between classes or types (grease is a subclass of lubricant); or between particulars and classes (this compressor is an instance of rotating equipment).

Extract from [SOURCE: ISO/IEC 21838-1] - minor edits to the example above to make them engineering focused.

3.5
data
representation of information in a formal manner suitable for communication, interpretation, or processing by human beings or computers

[SOURCE: ISO 10303-1:1994, 3.5.2] and [ISO 15926-2:2003]

3.6
information
facts, concepts, or instructions

[SOURCE: ISO 10303-1:1994(en), 3.2.20] and [ISO 15926-2:2003]

3.7
axiom
statement that is taken to be true, to serve as a premise for further reasoning

Note 1 to entry: Axioms may be formulated as natural language sentences or as formulae in a formal language. In the OWL community, 'Axiom' is used to refer to statements that say what is true in the domain that are 'basic' in the sense that they are not inferred from other statements.

[SOURCE: ISO 19150-1:2012] and [SOURCE: ISO/IEC 21838-1]

3.8
ontology
collection of terms, relational expressions and associated natural-language definitions together with one or more formal theories designed· to capture the intended interpretations of these definitions

[SOURCE: ISO/IEC 21838-1]

3.9
model
representation of certain entities and their characteristics and relationships either (a) using a formalism, or (b) using an established or ad hoc modelling paradigm, approach, or technique

[SOURCE: ISO 15704:2019] modified to add "and relationships"

3.10
system
combination of interacting elements organised to achieve one or more stated purposes

[SOURCE: ISO/IEC/IEEE 15939:2017]

3.11
use case
technique for capturing potential functional requirements

Note 1 to entry: This technique employs the use of one or more scenarios to convey how the system interacts with the end user or another system to achieve one or more specific objective.

Note 2 to entry: Typically use cases treat the system as a black box, and the interactions with the system, including system responses, are as perceived from the outside of the system. Use cases are popular because they simplify the description of requirements, and avoid the problem of making assumptions about how the functionality will be accomplished.

[SOURCE: IEC TS 62443-1-1:2009, 3.2.132]

3.12
primitive
expression for which no non-circular definition can be provided

[SOURCE: ISO/IEC 21838-2]

3.13
instance
individual that instantiates some class

EXAMPLES John, John's laptop, the year 2012.

3.14
expression
word or group of words or corresponding symbols that can be used in making an assertion

Note 1 to entry: Expressions are divided into natural language expressions and expressions in a formal language.

[SOURCE: ISO/IEC 21838-1]

3.15
term
expression that refers to some class or to some particular

Note 1 to entry: An ontology will typically contain a unique 'preferred term' for the entities within its coverage domain. Preferred terms may then be supplemented with other terms recognised by the ontology as synonyms of the preferred terms.

[SOURCE: ISO/IEC 21838-1]

3.16
definition
concise statement of the meaning of an expression

[SOURCE: adapted from ISO 15225:2016, 3.5]

Note 1 to entry: For the purposes of this standard, definitions can be of two sorts: (1) those formulated using a natural language such as English, supplemented where necessary by technical terms or codes used in some specialist domain; (2) those formulated using a computer-interpretable language such as OWL 2 or CL.

[SOURCE: ISO/IEC 21838-1]

3.17
Uniform Resource Identifier (URI)
A URI is an identifier for an entity in the universe of discourse, including physical things, documents, classes, numbers and strings.

Note 1 to entry. The referents of URIs in RDF data are called "resources".

[SOURCE: https://www.ietf.org/rfc/rfc3987.txt, https://www.w3.org/TR/rdf11-concepts/]

3.18
Resource description framework (RDF)
RDF is a standard model for data interchange on the Web recommended by the W3C.

[SOURCE: https://www.w3.org/RDF/]

3.19
RDF property
An RDF property is a specified relation between subject and object in a triple that imposes obligations (or restrictions) on the relation.

Note 1 to entry: W3C prescribes a list of RDF properties such as rdf:type and rdfs:subClassOf

Note 2 to entry: An RDF property may have sub-properties with restrictions on domain and range.

3.20
OWL 2
The OWL 2 Web Ontology Language [1], informally OWL 2, is an ontology language recommendation by the World Wide Web Consortium (W3C) with formally defined meaning, applying description logic semantics. OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents.

[SOURCE: https://www.w3.org/TR/owl2-syntax/]

3.21
OWL DL reasoning
Inference or implicit facts from knowledge expressed in an OWL ontology, using the inference rules of the description logic on which OWL 2 is based.

3.22
OWL Manchester Syntax
The Manchester syntax is a user-friendly compact syntax for OWL 2 ontologies; it is frame-based, as opposed to the axiom-based other syntaxes for OWL 2.

3.23
Domain and range
Domains and ranges are specified in the definitions of RDF properties in order to enable inferences about the things described using those properties. A domain is the class to which the subject of an RDF statement using a given property belongs, and a range is the class of its object (value).

[SOURCE: https://www.dublincore.org/resources/glossary/domains_and_ranges/]

3.24
Object property
A relation between individuals, part of the vocabulary of an OWL ontology. An object property is an RDF property.

3.25
Data property
A relation from an individual to a literal value, with semantic significance.

3.26
Annotation property
RDF property with no semantic significance, suitable for data that is not subject to OWL reasoning.

3.27
Prefix
Following the W3C notation for abbreviating namespaces. Examples of prefix include rdf: and owl:.

3.28
Reference data library
A reference data library is a vocabulary of industrial terms organised according to ontological categories.

3.29
Asset model
A collection of individuals, classified and related in accordance with a reference data library and, representing an industrial asset.

3.30
Pattern
A pattern provides the grammar for expressing facts.

Note 1 to entry: IDO provides a set of reusable patterns.

Note 2 to entry: Any use case will call for the introduction of additional patterns which are often derived from the patterns described in this document.

4 Abbreviated terms

RDF/OWL OWL represented in RDF

UML Unified Modeling Language

5 IDO Entities

5.1 Reading guide for this chapter

This chapter provides the contents of the IDO ontology as a readable document.

Section 5.2 describes metadata for the ontology. Blocks of code are in OWL Manchester Syntax.

Sections 5.3 to 5.6 describe the full contents of the IDO ontology in a readable form. Each sub-section covers one ontology resource (class or relation), with annotations (definition, cross-references, notes, etc.) and any semantic OWL restrictions. The names of OWL annotation properties for resources are abbreviated according to table 5.1. Semantic restrictions on resources are shown as blocks of code in OWL Manchester Syntax.

Table 5.1 — Labels of RDF annotation properties

annotation property	entry label
iof-av:isPrimitive	primitive?
iof-av:naturalLanguageDefinition	definition
iof-av:primitiveRationale	why primitive
iof-av:usageNote	usage note
owl:deprecated	deprecated?
rdfs:seeAlso	see also
skos:example	example
skos:scopeNote	scope note
skos:altLabel	alternative label
iof-av:explanatoryNote	explanatory note
rdfs:comment	comment
rdfs:isDefinedBy	defined by
lis:relatedEntityISO15926	related in ISO 15926
lis:equivalentEntityISO15926	equivalent in ISO 15926
lis:deprecatedEntityISO15926	deprecated ISO 15926
lis:remodelsEntityISO15926	remodels ISO 15926
iof-av:firstOrderLogicDefinition	first-order logic definition
iof-av:semiFormalNaturalLanguageDefinition	semi-formal definition
iof-av:semiFormalNaturalLanguageAxiom	semi-formal axiom
iof-av:adaptedFrom	adapted from
iof-av:synonym	synonym

Diagrams are provided to aid in navigation of the class and relation taxonomies, and to illustrate modelling examples in a visual form. They follow a UML *class diagram* style and represent the RDF/OWL representation of IDO as faithfully as is practical. Boxes represent RDF resources (the *subject* and *object* of RDF triples), and arrows represent relationships (the *predicate* of RDF triples). Following UML convention, arrows with open arrowheads indicate subclass relationships. Boxes are provided with circled letters to indicate OWL type of the relevant resource, as follows.

(C) OWL class

(O) OWL object property (individual to individual relation)

(D) OWL data property (individual to literal value relation)

(A) OWL annotation property (non-semantic annotation)

The diagrams are intended as an aid to the reader, not as a complete reproduction of the IDO ontology or of example models. Some contents of the ontology will be left out or truncated, to avoid overly complex diagrams. This applies in particular to axioms that provide semantic restrictions.

5.2 NOTE on "primitive" declarations (for the editors)

All classes and relations need to be marked "primitive" or "not primitive". However, precisely what this means remains to be clarified. **The current "primitive rationale" annotations are only indicative of what we need to provide for the standard.**

5.3 Declarations

5.3.1 Comment at top of file

```
##
## Industrial Data Ontology (IDO) work-in-progress, Q2 2023
## This ontology is the continuation of ISO 15926-14 (TR, 2020)
## This document is in OWL 2 Manchester Syntax, see https://www.w3.org/TR/owl2-manchester-
syntax/
##
```

5.3.2 Prefixes

```
## Prefixes
Prefix: lis: <http://rds.posccaesar.org/ontology/lis14/rdl/>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix: xml: <http://www.w3.org/XML/1998/namespace>
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix: skos: <http://www.w3.org/2004/02/skos/core#>
Prefix: pav: <http://purl.org/pav/>
Prefix: obo: <http://purl.obolibrary.org/obo/>
Prefix: dol: <http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl#>
Prefix: ssn: <http://www.w3.org/ns/ssn/>
Prefix: time: <http://www.w3.org/2006/time#>
Prefix: om: <http://www.ontology-of-units-of-measure.org/resource/om-2/>
Prefix: foaf: <http://xmlns.com/foaf/0.1/>
Prefix: dc: <http://purl.org/dc/elements/1.1/>
Prefix: dcterms: <http://purl.org/dc/terms/>
Prefix: dcmitype: <http://purl.org/dc/dcmitype/>
Prefix: prov: <http://www.w3.org/ns/prov#>
Prefix: lis2: <http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003#>
Prefix: lis12: <http://standards.iso.org/iso/15926/ontology/life-cycle-integration/>
Prefix: iof-av:
<https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/>
```

5.3.3 Ontology

```
## Ontology declaration
Ontology: <http://rds.posccaesar.org/ontology/lis14/ont/core>
<http://rds.posccaesar.org/ontology/lis14/ont/core/1.1-alpha.0>
     Import:
<https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/>
```



```
    Annotations: rdfs:label "Industrial Data Ontology",
      dcterms:title "Industrial Data Ontology",
      owl:versionInfo "1.1-alpha.0 WIP 2023-05-10",
      pav:derivedFrom <http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003>,
      # omit pav:previousVersion <http://standards.iso.org/iso/15926/part14/1.1>,
      pav:previousVersion <http://rds.posccaesar.org/ontology/lis14/ont/core/1.0>,
      pav:lastUpdateOn "2023-05-10T11:55:36Z"^^xsd:dateTime ,
      rdfs:seeAlso <http://rds.posccaesar.org/ontology/lis14/ont/annotationsPart2>,
      dcterms:license <https://creativecommons.org/licenses/by-sa/4.0/>,
      dcterms:creator <https://orcid.org/0000-0002-7167-7321>,
      dcterms:contributor <https://orcid.org/0000-0002-7167-7321>,
      dcterms:contributor <https://orcid.org/0000-0002-6228-1317>,
      dcterms:contributor <https://orcid.org/0000-0002-5509-8899>,
      dcterms:contributor <https://orcid.org/0000-0002-9736-8316>,
      # not issued yet dcterms:issued "2021-08-06"^^xsd:date,
      dcterms:modified "2023-05-10"^^xsd:date,
      dcterms:publisher <https://www.posccaesar.org>,
      dc:rights "Copyright POSC Caesar Association",
      rdfs:comment "The Industrial Data Ontology is an upper ontology for industrial da-
ta, inspired by the ISO 15926-2 data model and Basic Formal Ontology.",
      dcterms:description "The Industrial Data Ontology is an upper ontology for indus-
trial data, inspired by the ISO 15926-2 data model and Basic Formal Ontology."
      # foaf:isPrimaryTopicOf <provide link to documentation here later>
```

NOTE. The namespace URI of the IDO ontology and contained entities is http://rds.posccaesar.org/ontology/lis14/, referring to an organisation external to ISO. This is in line with common practice. For example, the OWL rendering of the BFO-ISO ontology of ISO 21838-2 employs the namespace http://purl.obolibrary.org/obo/.

5.4 Classes

```
## Class details

## Disjointness clause for top classes:
DisjointClasses: lis:Temporal, lis:Object, lis:Dependent
```

5.4.1 lis:Object

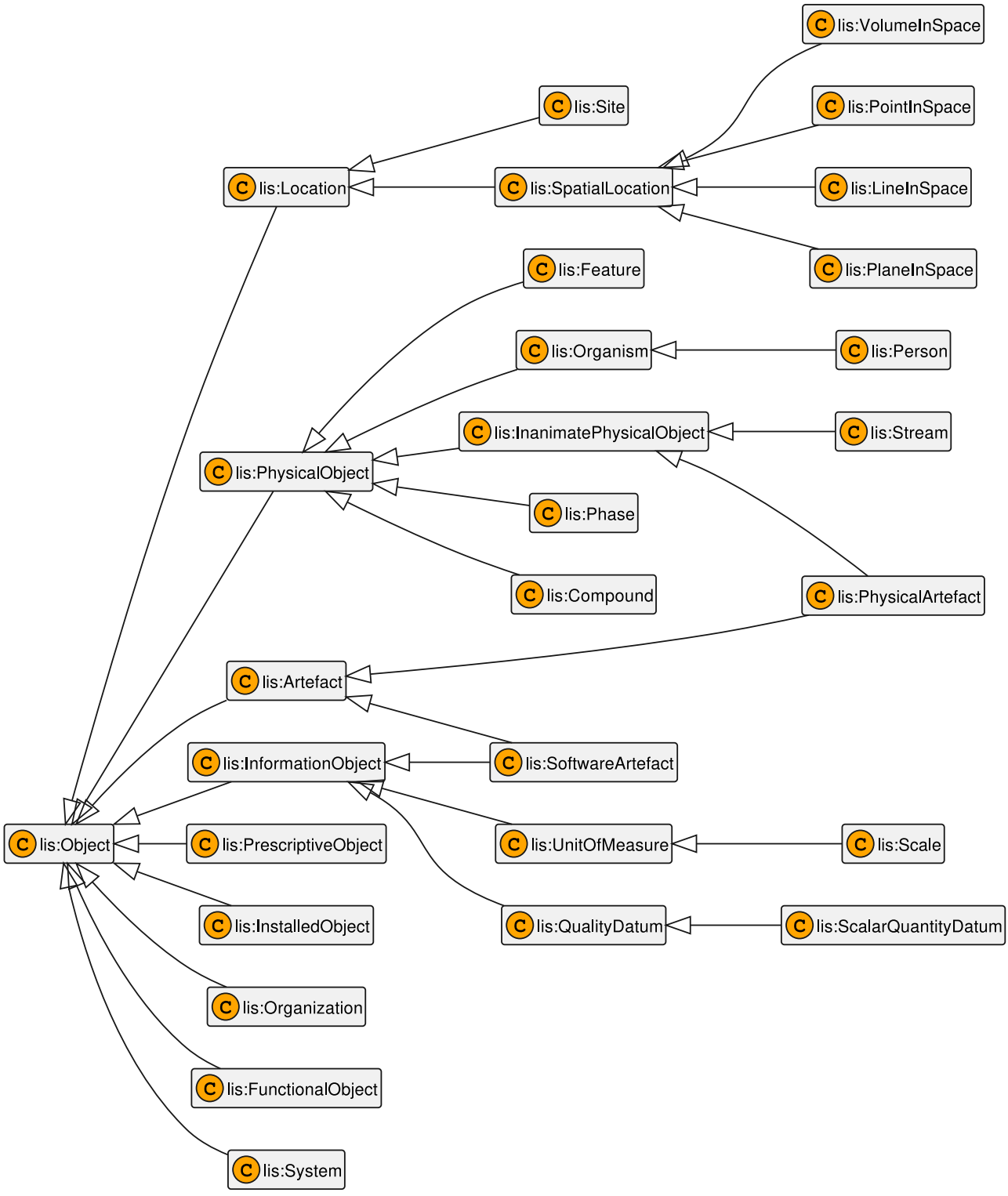


Figure 5.1 — IDO 'object' class hierarchy

definition An entity that is not dependent on other entities for existence, and which persists through time.

primitive? true

why primitive A precise account of what counts as an 'object' is beyond the scope of IDO.

example A process plant, a compressor, a wire, an operator, a database record, an industry standard.

see also dol:endurant

comment From the DOLCE description of 'endurant': "The main characteristic of endurants is that all of them are independent essential wholes. …"

related in ISO 15926 lis2:PossibleIndividual

related in ISO 15926 lis12:Individual

see also obo:BFO_0000004

comment BFO_0000004 is 'independent continuant'.

scope note In IDO, 'information object' is represented as a subclass of 'object'. According to the BFO literature, and applied in the Information Artifact Ontology and other BFO compliant ontologies, information artifacts are (multiply) dependent objects. a *sibling* class to 'independent continuant' rather than a subclass, as used in IDO.

```
DisjointClasses: lis:PhysicalObject, lis:InformationObject, lis:Organization, lis:Location
```

5.4.1.1 lis:Artefact

NOTE. Tentative addition.

definition An object that is made by humans.

primitive? true

why primitive To give a precise account of "making" goes beyond the scope of IDO. An account would engage notions on which there is no clear consensus, including intentional notions of intended purpose, and of goal-directed action; of permanence and integrity; etc. There is an abundance of edge cases for "artefact".

example Among physical objects, a pressure transmitter, a process plant, a wrench, a computer; among information objects, a diagram, a PDF document, a software program, a binary file.

EXAMPLE. The 'instrumentation technician' John makes a 'pressure gauge assembly'. This can be expressed in OWL with the following axioms and diagram.

```
Class: ex:InstrumentationTechnician
  SubClassOf: lis:Person
Class: ex:PressureGaugeAssembly
  SubClassOf: lis:Artefact
ObjectProperty: ex:makes
Individual: ex:P-12345-PA-01
  Types: ex:PressureGaugeAssembly
Individual: ex:John
  Types: ex:InstrumentationTechnician
  Facts: ex:makes ex:P-12345-PA-01
```

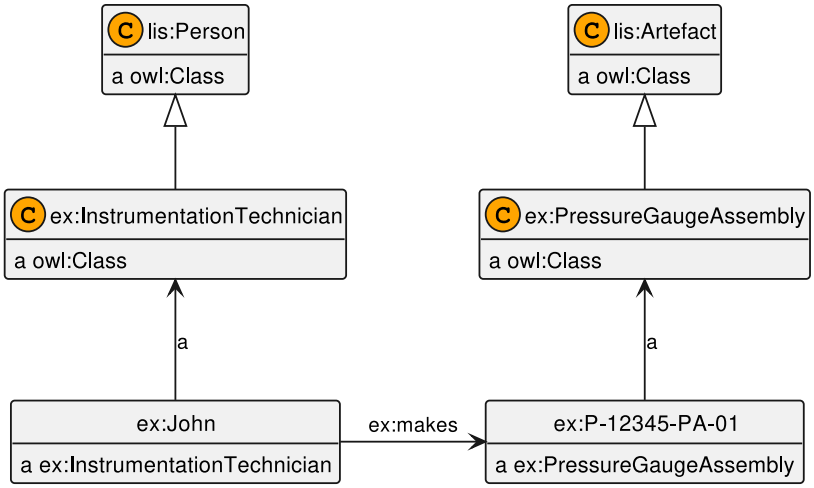



Figure 5.2 — 'artefact' example, simple pattern

For a generic 'makes' relation, a domain of 'person' or 'organization' and a range of 'artefact' is natural.

```
ObjectProperty: ex:makes
  Domain: lis:Organization or lis:Person
  Range: lis:Artefact
```

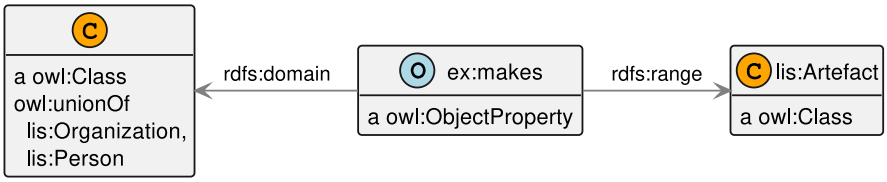


Figure 5.3 — 'artefact' example, domain and range for 'makes' example relation

A more detailed model is possible, with explicit 'role' individuals; then the 'makes' relation appears as a shortcut over the pattern of roles and created output of activities, as in the following. See the example accompanying lis:Role for details on 'maker role'.

```
Class: ex:Assembling
Class: ex:InstrumentMakerRole
Individual: ex:MakingP12345-PA-02
  Types: ex:Assembling
  Facts: lis:creates ex:P-12345-PA-01
Individual: ex:JohnsInstrumentMakerRole
  Types: ex:InstrumentMakerRole
  Facts: lis:realizedIn ex:MakingP12345-PA-02
Individual: ex:John
  Facts: ex:makes ex:P-12345-PA-01, ex:hasMakerRole ex:JohnsInstrumentMakerRole
```

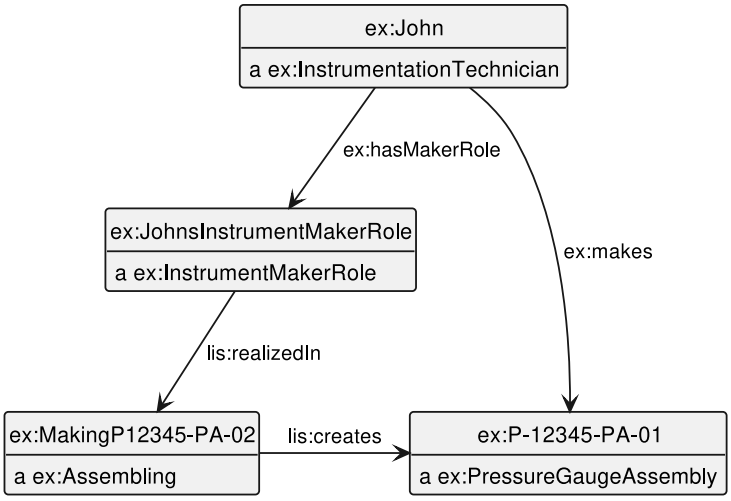


Figure 5.4 — 'artefact' example, pattern with role

The detailed model motivates adding a property chain to ensure 'makes' relationships are indeed inferred from the detailed pattern.

```
ObjectProperty: ex:hasMakerRole
ObjectProperty: ex:makes
  SubPropertyChain: ex:hasMakerRole o lis:realizedIn o lis:creates
```

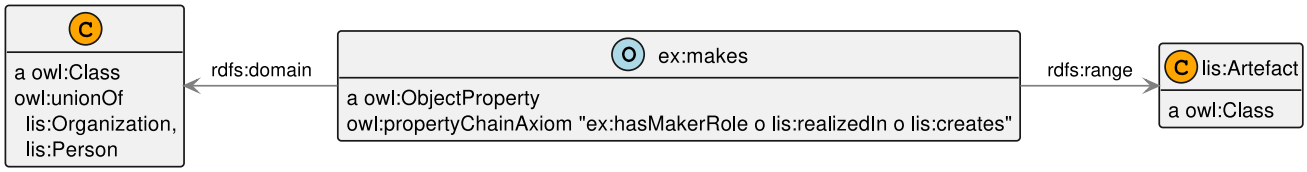


Figure 5.5 — 'artefact' example, property path for 'makes' example relation

5.4.1.2 lis:PhysicalArtefact

definition An 'artefact' that is a 'physical object'.

primitive? false

semi-formal axiom If x is a 'physical artefact', then x is an 'artefact' and a 'physical object'.

example A worktable, a bolt, a digital display unit, a pressure vessel.

comment The PCA RDS defines RDS422594 'artefact' as, "An inanimate physical object that is made or given shape by man."

```
Class: lis:PhysicalArtefact
  SubClassOf: lis:InanimatePhysicalObject
```

5.4.1.3 lis:SoftwareArtefact

definition An item that is made during the software development process.

primitive? true

why primitive The notion of software development is not defined.

example A CAD application, a SQL query, a backup script.

see also obo:IAO_0000010

explanatory note IAO_0000010 is 'software'. The Information Artifact Ontology includes the following subclasses: 'software application', 'software library', 'software method', 'software module', 'software script'.

```
Class: lis:SoftwareArtefact
  SubClassOf: lis:InformationObject
```

5.4.1.4 lis:System

definition An 'object' with at least one function that can only be realized given that the functions of one or more parts are realized as well.

primitive? true

why primitive A precise account of the nature of dependency between the function of a system, as a whole, and the functions of its parts is (likely) beyond the scope of IDO.

example A pressure vessel, a crane, a pipeline.

comment TODO consider "accidental" dependency of system function on parts, versus "essential", with a composition relation between system function and the functions of parts.

see also lis:functionalPartOf

see also ssn:System

comment The SSN definition of 'System' (2017): "System is a unit of abstraction for pieces of infrastructure that implement Procedures. A System may have components, its subsystems, which are other Systems."

```
Class: lis:System
  SubClassOf: lis:hasFunctionalPart some lis:FunctionalObject
```

5.4.1.5 lis:FunctionalObject

definition An 'object' that has a 'function'.

primitive? false

semi-formal axiom If x is a 'functional object', then x 'has function' some 'function'.

example A pump, a hammer, a stairwell, a pipeline.

deprecated? true

usage note Artefact types tend to be described primarily in terms of 'function'; for example, the "minimal" definition of a pump would be, "a physical artefact, the function of which is to pump (to be the active participant in a pumping activity)". This kind of definition places no restriction on physical make-up and is suitable for a subclass of 'functional object'. Note that the activity type of 'pumping' is here a precursor to the definition of the physical object type 'pump'. A taxonomy of pump classes should extend from a class 'pump' defined solely in terms of function, as above.

related in ISO 15926 lis2:FunctionalPhysicalObject

comment ISO 15926-2, p. 110, states, "A functional_physical_object is a physical_object that has functional, rather than material, continuity as its basis for identity. Adjacent temporal parts of a functional_physical_object need not have common matter or energy, provided the matter or energy of each temporal part fulfils the same function."

related in ISO 15926 lis12:FunctionalPhysicalObject

related in ISO 15926 lis12:ClassOfPhysicalObjectByFunction

```
Class: lis:FunctionalObject
  SubClassOf: lis:hasFunction some lis:Function
```

5.4.1.6 lis:PhysicalObject

definition An entity that has some material part.

primitive? true

why primitive A precise account of what it means to be a physical object is beyond the scope of IDO. There is no obvious consensus on the notion of "material".

example A wall, a fluid stream, a welder, a printed manual, a seafloor.

equivalent in ISO 15926 lis2:PhysicalObject

comment IDO asserts that 'physical object' and 'activity' are disjunct classes. In contrast to this, and according to a 4D approach, ISO 15926-2, p. 26, states that "physical_object and activity are not mutually exclusive. A possible_individual can be both".

equivalent in ISO 15926 lis12:PhysicalObject

deprecated ISO 15926 lis2:MaterializedPhysicalObject

deprecated ISO 15926 lis12:MaterializedPhysicalObject

see also obo:BFO_0000040

explanatory note BFO_0000040 is 'material entity'.

see also dol:physical-object

5.4.1.7 lis:InanimatePhysicalObject

definition A 'physical object' that is not living.

primitive? true

why primitive An account of what "living" means is beyond the scope of IDO.

example A tool, a building, a vehicle, an instrument, a parking lot.

related in ISO 15926 lis2:ClassOfInanimatePhysicalObject

comment ISO 15926-2, p. 122: "A class_of_inanimate_physical_object is a class_of_arranged_individual whose members are not living."

equivalent in ISO 15926 lis12:InanimatePhysicalObject

5.4.1.8 lis:Stream

NOTE. The definition here is obviously a tentative attempt.

definition An 'inanimate physical object', consisting of energy or homogeneous parts, with velocity along a path determined by a 'site'.

primitive? true

why primitive An full account of the criteria for what counts as a stream is beyond the scope of IDO.

example A stream of fluid in a pipeline, electricity in an electrical wire, ore moving on a conveyor belt.

usage note The class 'stream' is intended to be used for flows of material or energy, in accordance with conventional use. There is an abundance of cases where use of the term "stream" might be called into question, as in, is a stream of electricity really moving; how many cars are needed to participate before it is properly called a "flow" of cars; while a flow of sand is a stream, maybe a succession of rolling rocks would not qualify.

equivalent in ISO 15926 lis2:Stream

comment ISO 15926-2, p. 112, states, "A stream is a physical_object that is material or energy moving along a path, where the path is the basis of identity and may be constrained. The stream consists of the temporal parts of those things that are in the stream whilst they are in it."

equivalent in ISO 15926 lis12:Stream

5.4.1.9 lis:Compound

definition A physical object with a 'quality' representing material composition.

primitive? true

why primitive The semantics of materials, of which compounds would be a specialisation, is beyond the scope of IDO.

example A volume of complex hydrocarbons, a steel rod, a foam insulation panel; a subclass of compound would in each case contribute the material characterisation.

scope note Natural kinds are expressly not to be classified as 'compound'. The class 'compound' is only to be used to classify amounts of matter, with non-semantic auxiliaries used to refer to chemical composition. The semantics of natural kinds such as "helium", "steel", or "hydrocarbon" is out of scope for IDO.

usage note For explicit meaning of a 'compound' subclass, such as "steel [object]", employ an RDL 'quality' class representing "material composition", with restrictions to a modelling construct representing steel.

usage note The class 'compound' serves as an ultimate "book-keeping" superclass for a taxonomy of material make-up. It has been found useful in managing classifications (of physical objects) according to material composition. It carries no distinguishing constraints, since any physical object will have material composition. The class 'compound' would perhaps better be named "physical object classified according to material composition".

related in ISO 15926 lis2:ClassOfCompound

equivalent in ISO 15926 lis12:Compound

see also https://rds.posccaesar.org/doc/patterns/pattern_stream_component/

comment The "Stream Components" modelling pattern illustrates how a taxonomy of chemical compounds may be used to characterise the 'quality' of material composition.

```
Class: lis:Compound
  SubClassOf: lis:hasQuality some lis:MaterialCompositionQuality
```

5.4.1.10 lis:Feature

definition A contiguous, non-separable part of a physical object.

primitive? true

why primitive A precise account of what "non-separable" means is beyond the scope of IDO.

example A flange face, the inside surface of a pump, a painted area of a manifold, a bolt threading.

related in ISO 15926 lis2:ClassOfFeature

equivalent in ISO 15926 lis12:Feature

see also obo:BFO_0000024

explanatory note BFO_0000024 is 'fiat object part'. This is at least very close in meaning to 'feature' as defined in in ISO 15926.

see also dol:relevant-part

EXAMPLE. A piping engineering project builds a catalogue of piping bulk items. The items are characterised by generic type (pipe, fitting, flange), material (alloys), and geometrical features, as specified in relevant industry standards. The designation "Weld Neck Flange NPS 2 CL150 RF WT2.77 ASTM 182 Gr F55" is an example of a fully specified piping flange:

"Weld Neck Flange": This term indicates that the flange is designed to be welded to a pipe or other component.

"NPS 2": This term specifies the nominal pipe size (NPS) of the flange, the inner diameter of the pipe to which it will be welded, according to ASME B36.10

"CL150": This term indicates the flange's pressure class rating, according to ASME B16.5

"RF": This term stands for "raised face," which is a type of flange face that provides a seal between two flanges, according to ASME B16.5

"WT2.77": This term indicates the wall thickness of the flange, according to ASME B36.10

"ASTM 182 Gr F55": This term indicates the material specification for the flange, according to ASTM 182

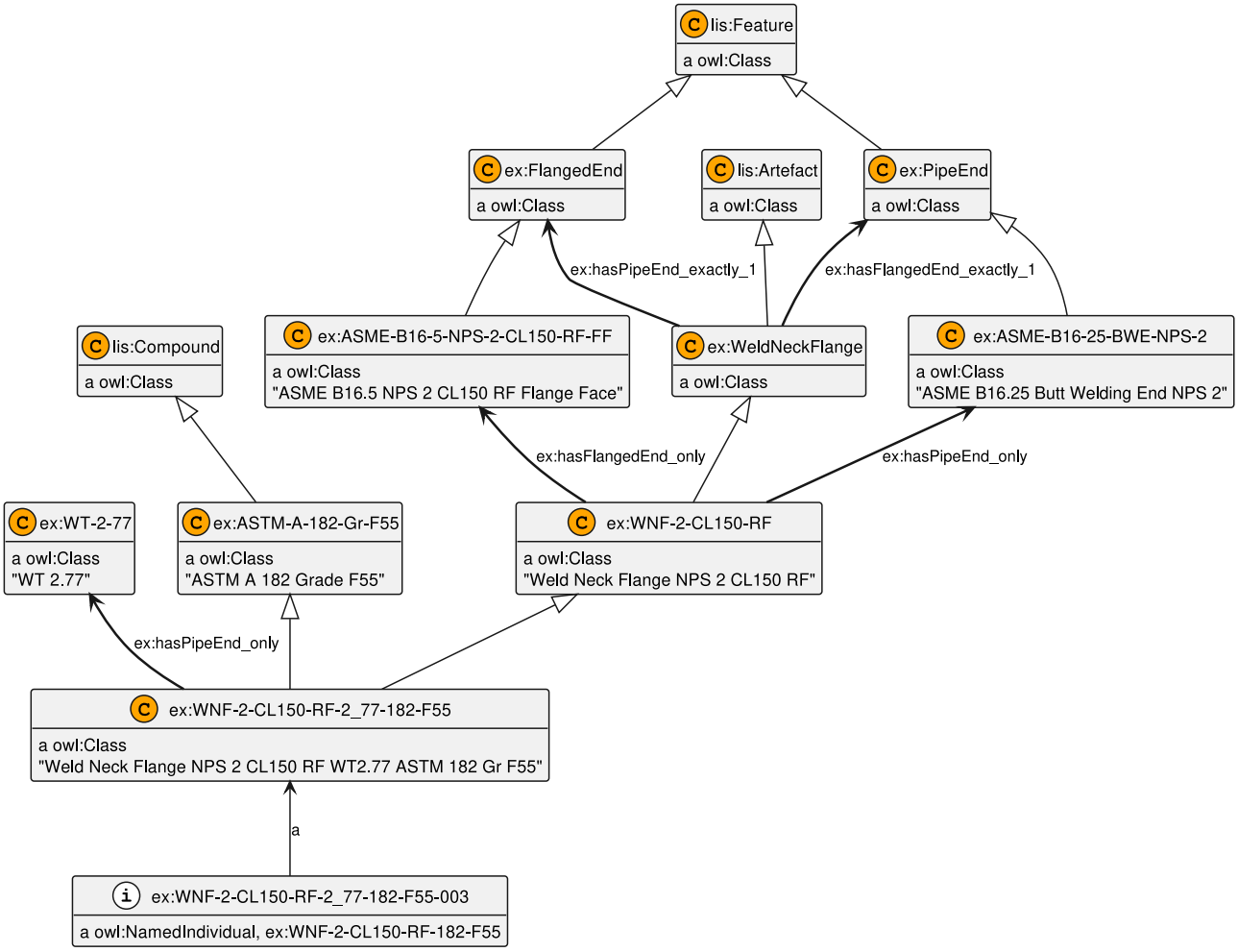


Figure 5.6 — 'feature' example, bulk material catalogue item

5.4.1.11 lis:Phase

definition A physical object with a 'quality' representing a uniformity of physical properties.

primitive? true

why primitive The semantics of phases is beyond the scope of IDO.

example A solid object, a liquid object, a gaseous object.

usage note The class 'phase' serves as an ultimate "book-keeping" superclass for a taxonomy of phases. It carries no distinguishing constraints, since any physical object will have phase.

related in ISO 15926 lis2:Phase

comment ISO 15926-2, p. 125: "A phase is a class_of_arranged_individual based on the nature of the boundary behaviour of material resulting from its atomic and molecular bonding."

related in ISO 15926 lis12:Phase

see also https://en.wikipedia.org/wiki/Phase_(matter)

comment "A simple description is that a phase is a region of material that is chemically uniform, physically distinct, and (often) mechanically separable. In a system consisting of ice and water in a glass jar, the ice cubes are one phase, the water is a second phase, and the humid air is a third phase over the ice and water. The glass of the jar is another separate phase." (Wikipedia)

```
Class: lis:Phase
  SubClassOf: lis:hasQuality some lis:PhaseQuality
```

5.4.1.12 lis:Organism

definition A living 'physical object'.

primitive? true

why primitive Providing precise semantics of what "living thing" amounts to is beyond the scope of IDO.

example Animal, plant, fungus, bacterium.

related in ISO 15926 lis2:ClassOfOrganism

equivalent in ISO 15926 lis12:Organism

5.4.1.13 lis:Person

definition A human 'organism'.

primitive? true

why primitive Providing precise semantics of what "human" amounts to is beyond the scope of IDO.

example A person, a child, an engineer.

scope note In industry use cases, subclasses of 'person' will typically require role attributions to be considered fully explicit. For example, 'maintenance engineer' is an obvious subclass of 'person', as every engineer is a person; use a 'role' to say what a maintenance engineer does. I.e., such a class may be augmented with a restriction to say that any member is the bearer of an engineering role; what the role amounts to will then be captured by the kinds of activities that count as realizations of the role.

related in ISO 15926 lis2:ClassOfPerson

equivalent in ISO 15926 lis12:Person

5.4.1.14 lis:InformationObject

TODO. Consider whether IDO should add a class to correspond to the more general 'generically dependent continuant'.

definition A pattern that 'is about' something else, and 'concretized as' at least one physical entity.

primitive? true

why primitive The definition implicitly calls on the BFO notion of generic dependence, as captured in BFO's 'generically dependent continuant', for which IDO doesn't have an analogue.

example A report, a drawing, a P&ID diagram, a purchase order, a certificate, a database, a database record, a measurement datum, a unit of measure, a SKOS vocabulary.

usage note A common application of 'information object' is in representing facts about the vocabulary of the ontology itself. The OWL facility of "punning" treats OWL classes and properties as OWL individuals, enabling OWL object properties to be applied to them. In use cases that involve punning, the SKOS vocabulary is recommended to be used: the SKOS notions of 'concept', 'concept scheme', and 'collection' should be represented as subclasses of 'information object'.

see also https://www.w3.org/TR/2012/REC-owl2-primer-20121211/#Entity_Declarations

scope note The BFO class 'generically dependent continuant' represents the notion of a pattern that exists in at least one copy, e.g., "the sequence of this protein molecule". This is more general than 'information object', and not at present included in IDO. 'information object' is a subclass, with the added condition that the pattern 'is about' something.

see also obo:IAO_0000030

comment IAO_0000030 is 'information content entity', a subclass of BFO_0000031 'generically dependent continuant' defined in the Information Artifact Ontology.

see also obo:BFO_0000031

comment BFO_0000031 is 'generically dependent continuant'.

related in ISO 15926 lis2:ClassOfInformationObject

related in ISO 15926 lis12:InformationObject

related in ISO 15926 lis12:InformationContent

5.4.1.15 lis:QualityDatum

definition An 'information object' that represents the magnitude of a 'quality'.

primitive? true

why primitive A precise account of the notion of representation of magnitudes is beyond scope of IDO.

example 10 volts, high (of pressure), low (of temperature), category 3 (as used to rate the strength 'quality' of a storm), 3 MB (a file size).

usage note This class may be used for data that don't fit the restriction to "single unit, single number" of 'scalar quantity datum', including data employing vectors instead of numbers, and conventional categories like "high" and "medium".

scope note Introduce subclasses of 'quality datum' in RDL to distinguish categories of data, such as nominal versus measured values.

comment This class is inspired by the class "measurement datum" of the Information Artefact Ontology. The change of wording from "measurement" to "quality" is intended to support cases where measurement is not involved, such as with nominal values.

see also obo:IAO_0000109

comment IAO_0000109 is 'measurement datum', a class in the Information Artifact Ontology.

related in ISO 15926 lis12:Quantity

```
Class: lis:QualityDatum
  SubClassOf: lis:quantifiesQuality some lis:Quality
```

5.4.1.16 lis:ScalarQuantityDatum

TODO. The datumValue restriction here is to rdfs:Literal, but it would be better to require a generic numeric type.

definition A 'quality datum' that represents the magnitude of a 'physical quantity' with a single unit of measure and a single numeric value.

primitive? false

semi-formal axiom If x is a 'scalar quantity datum', then x is a 'quality datum' which 'quantifies quality' some 'physical quantity', and has a 'datum uom' relationship to a 'unit of measure' and a 'datum value' relationship to a literal value (actually, to a number).

example 10 kilograms, 220 volt.

usage note A scalar quantity datum has a unique unit of measure and a unique numeric value. This class is inspired by the class "scalar measurement datum" of the Information Artefact Ontology.

see also obo:IAO_0000032

comment IAO_0000032 is 'scalar measurement datum', a class in the Information Artifact Ontology.

see also om:Measure

comment OM 2 describes 'measure' as follows. "A measure combines a number to a unit of measure. For example, "3 m" is a measure."

see also http://qudt.org/schema/qudt/QuantityValue

comment QUDT describes 'quantity value' as follows. "A Quantity Value expresses the magnitude and kind of a quantity and is given by the product of a numerical value n and a unit of measure U. The number multiplying the unit is referred to as the numerical value of the quantity expressed in that unit."

related in ISO 15926 lis12:PhysicalQuantity

```
Class: lis:ScalarQuantityDatum
  SubClassOf: lis:datumUOM some lis:UnitOfMeasure, lis:datumValue some rdfs:Literal
```

5.4.1.17 lis:UnitOfMeasure

definition A definite magnitude of a 'quality', defined and adopted by convention or by law, that is used as a standard for measurement of the same kind of 'quality'. (rephrased from Wikipedia)

primitive? true

why primitive Precisely determining what qualifies as a unit of measure is beyond the scope of IDO.

example centimetre, kilogram per minute, Celsius, lunar distance, kilo-lines of code

related in ISO 15926 lis12:QuantityMeasure

see also om:Unit

comment 'unit' is a class in the Ontology of units of Measure (OM 2).

see also obo:UO_0000000

comment UO_0000000 is 'unit', a class in the OBO Units of measure ontology.

5.4.1.18 lis:Scale

definition A 'unit of measure' for which the range of numbers is determinate.

primitive? true

why primitive A comprehensive definition of the notion of scale is beyond the scope of IDO.

example centimetre, joule per kelvin; nominal, ordinal, interval, and ratio scales.

scope note Arguably, IDO conflates the notions of "unit of measure" and "scale" in an unhealthy way. While a unit of measure is a definite magnitude, a scale arises from fixing a number range to express multiples of the unit of measure. One may argue that a unit of measure has an implicit scale, obtained by multiplying the unit by any number; this will then allow for undefined or meaningless expressions like "minus 10 Kelvin". The term "Scale" is preserved from ISO 15926-2, which doesn't make use of "unit of measure".

usage note The permissible number range for a scale may be expressed in OWL. Semi-formal example: If 'Celsius' is the 'uom of datum' of x, then any 'datum value' of x is within the range of -273,1 to infinity.

see also http://qudt.org/schema/qudt/Scale

comment QUDT provides the following subclasses of 'scale': 'Enumeration scale', 'Interval scale', 'Nominal scale', 'Ordinal scale', 'Ratio scale'. Note that the notion of nominal scales diverges in being a coding scheme, with numbers not implying relative magnitude.

see also om:Scale

equivalent in ISO 15926 lis2:Scale

equivalent in ISO 15926 lis12:Scale

5.4.1.19 lis:Organization

definition An aggregate of one or more ‘person’s that ‘has role’ some ‘role’.

primitive? true

why primitive A comprehensive definition of the notion of organization is beyond the scope of IDO (it would determine which kinds of aggregations of people, and which kinds of roles, are appropriate for organizations).

example A standards body, a a service team, a workgroup

related in ISO 15926 lis2:ClassOfOrganization

equivalent in ISO 15926 lis12:Organization

5.4.1.20 lis:Location

definition A region in space.

primitive? true

why primitive A comprehensive definition of the notion of location is beyond the scope of IDO.

example An outdoor plant area, an opening in a wall, the space inside an enclosure

equivalent in ISO 15926 lis2:SpatialLocation

equivalent in ISO 15926 lis12:RegionInSpace

see also dol:space-region

5.4.1.21 lis:Site

definition A three-dimensional ‘location’, demarcated by material entities serving as boundaries or points of reference.

primitive? true

why primitive A comprehensive definition of the notion of demarcation is beyond the scope of IDO.

example An office, a flight deck, the inside of a manifold, the hold of a ship.

scope note The position of a ‘site’ follows the position of the ‘physical object’ by which it is demarcated.

scope note Locations such as countries and cities are also considered sites.

usage note The utility of a ‘site’ may be represented by assigning one or more ‘role’s, such as ‘front desk role’, ‘storage space role’, or ‘entryway role’.

comment This class is inspired by the BFO/Information Artefact Ontology class BFO_0000029 ‘site’.

see also obo:BFO_0000029

comment BFO_0000029 is ‘site’

related in ISO 15926 lis2:SpatialLocation

related in ISO 15926 lis2:RegionInSpace

5.4.1.22 lis:SpatialLocation

TODO. Discuss in the community whether we need plural versions of the subclasses – such could be included, with naming like “point in space region” or “region of point in space” and accordingly for “line”, “plane”, and “volume”. If there’s a need for this, we can extend the vocabulary.

NOTE. The class ‘volume in space’ is a renaming of the former ‘region in space’, as this is better aligned with the vocabulary of (spatio-) temporal regions.

definition A determinate region in space.

primitive? true

why primitive To provide an account of space, or frame of reference, is beyond the scope of IDO.

example The cube-shaped region extending 1 metre in each direction from origo, the line extending 10 metres from origo along the z axis, two points with coordinates on a two-dimensional plane; the location of a container at 10:00 AM.

usage note The extent of a 'spatial location' may be given either in terms of a spatial frame of reference, for instance with coordinates, or by position relative to material entities at a point in time.

usage note As with 'spatiotemporal region', a spatial location need not be connected; for example, a location may consist in several cube-shaped regions.

usage note The subclasses 'point in space', 'line in space', 'plane in space', and 'volume in space' are all defined as singular: for example, a 'point in space' individual may not have multiple points as parts. This, in contrast to the analogous BFO classes 'one-dimensional spatial region', etc., which are all defined as plural, allowing for disconnected parts.

scope note The position of a 'spatial location' is fixed (relative to the frame of reference); it is not dependent on changing positions of 'physical object's.

see also obo:BFO_0000006

comment BFO_0000006 is 'spatial region'.

related in ISO 15926 lis2:SpatialLocation

related in ISO 15926 lis2:RegionInSpace

5.4.1.23 lis:PointInSpace

definition A zero-dimensional 'spatial location' with no proper parts.

primitive? true

why primitive A comprehensive account of geometries is beyond the scope of IDO.

example The origo, the location of a probe at a given point in time

equivalent in ISO 15926 lis12:PointInSpace

5.4.1.24 lis:LineInSpace

NOTE. Tentative addition [2023-02-17; ISO 15926-2 doesn't have a corresponding entity type.]

definition A one-dimensional, connected 'spatial location'.

primitive? true

why primitive A comprehensive account of geometries is beyond the scope of IDO.

example A line connecting two points with coordinates relative to the frame of reference, the path of a flexible pipe at 12:00

equivalent in ISO 15926 lis12:PointInSpace

5.4.1.25 lis:PlaneInSpace

NOTE. Tentative addition [2023-02-17; ISO 15926-2 doesn't have a corresponding entity type.]

definition A two-dimensional, connected 'spatial location'.

primitive? true

why primitive A comprehensive account of geometries is beyond the scope of IDO.

example The midway horizontal plane of a volume given by coordinates, the plane of a moving work platform at 12:00

equivalent in ISO 15926 lis12:PointInSpace

5.4.1.26 lis:VolumeInSpace

NOTE. This has been renamed from “RegionInSpace” to better align with the use of “region” in the ‘temporal’ taxonomy.

definition A three-dimensional, connected ‘spatial location’.

primitive? true

why primitive A comprehensive account of geometries is beyond the scope of IDO.

example A volume designated for 3D engineering at a point in time, a box-shaped volume determined by coordinates in a 3D frame of reference; the extent of a gas leak at 12:00

equivalent in ISO 15926 lis12:RegionInSpace

5.4.2 lis:Dependent

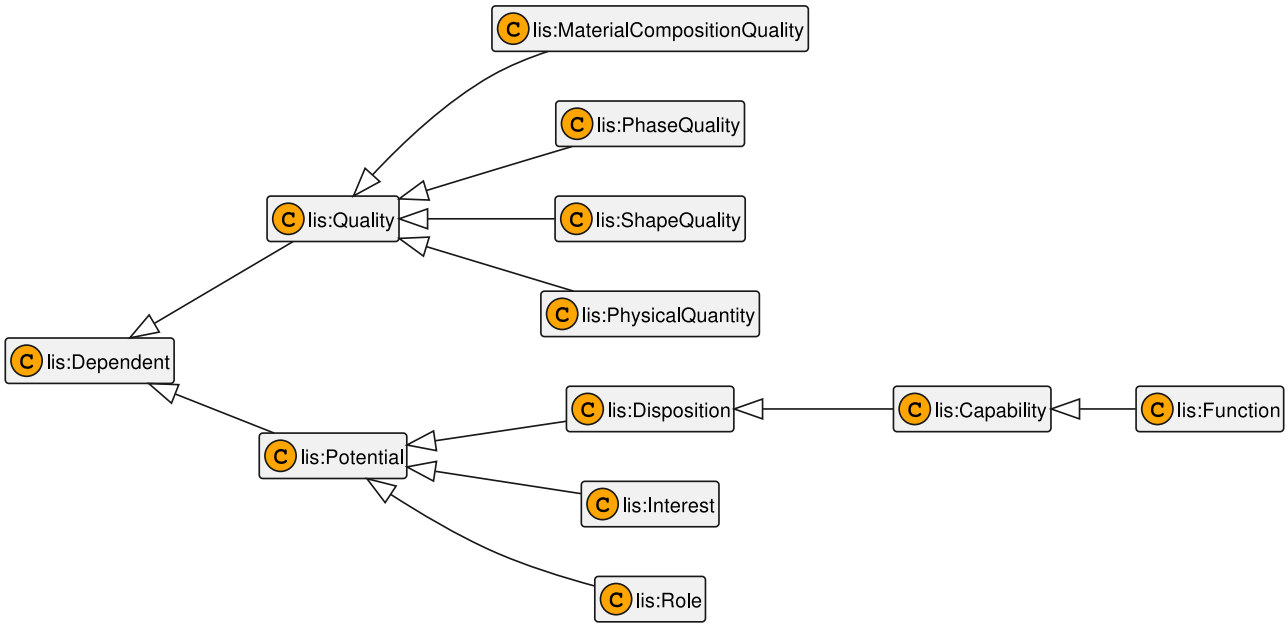


Figure 5.7 — IDO ‘dependent’ class hierarchy

alternative label Dependent Entity

alternative label Aspect

comment “Dependent” is a candidate better name than "Aspect" of ISO 15926-14, 2023-03-20.

definition An individual that exists only as dependent on an ‘object’ bearer. ‘dependent’ individuals enable the ontology to characterise individuals by properties they bear. This is divided into (i) inherent properties (‘quality’) like colour and mass, and (ii) properties that express potential (‘potential’), realized only through participation in activities, like the function of illuminating a space or the role of serving as a boundary.

primitive? true

why primitive To provide necessary and sufficient conditions is beyond scope of IDO. Note that the BFO analogue, ‘specifically dependent continuant’, is defined in terms of the primitive relation ‘specifically depends on’ (s-depends on), which is not included in IDO.

example The colour of a surface, the temperature of a medium, the storage-space role of a room.

scope note Any ‘object’ will be the bearer of various ‘dependent’ entities, but any ‘temporal’ entity has only an indirect relationship to ‘dependent’s. This is in line with BFO, but also a common source of confusion. A common need is to relate a measure based on a ‘physical quantity’ to an ‘activity’, like average speed to a transport activity; this may not be implemented as an “average speed” dependent entity of the transport. For a suitable representation pattern, see the class ‘activity profile’ and related resources.

see also lis:ActivityProfile

comment The class Dependent is closely related to "specifically dependent continuant" in BFO (BFO_0000020).

adapted from obo:BFO_0000020

comment BFO_0000020 is 'specifically dependent continuant'.

comment There is no direct analogue to 'dependent' entities in ISO 15926-2 (or in ISO 15926-12). However, lis2:Property and lis2:FunctionalPhysicalObject largely serve the same purpose as, respectively, 'quality' and 'potential'.

related in ISO 15926 lis2:Property

related in ISO 15926 lis2:FunctionalPhysicalObject

```
DisjointClasses: lis:Potential, lis:Quality
```

5.4.2.1 lis:Potential

alternative label RealizableEntity

comment "Potential" is a candidate better name than "Realizable Entity" of ISO 15926-14, 2023-03-20.

definition A dependent entity that represents a potential of its bearer. Realization is specified by a type of activity, and appropriate kind of participation: a necessary condition for realization is that the bearer of the potential participates, in the way specified, in an activity of the specified kind.

primitive? true

why primitive The necessary condition for being a 'potential' is that activity type and participation type are provided. It is however not clear what sufficient conditions would consist in, as these would at least involve a notion of successful execution for the relevant activities, which goes beyond the scope of IDO.

example The storage-space role of a room, the sealing function of a gasket, the disposition to corrode of a steel bar.

comment Use OWL property chains to enforce constraints on how the bearer needs to participate in order for an activity to count as a realization.

comment Inspired by the BFO class 'realizable entity' (BFO_0000017).

see also obo:BFO_0000017

comment BFO_0000017 is 'realizable entity'.

see also https://www.nature.com/articles/npre.2008.1941.1

comment URI of the paper Arp and Smith, 2008, "Function, Role, and Disposition in Basic Formal Ontology".

5.4.2.2 lis:Disposition

definition A 'potential' for which realization depends on the physical make-up of the bearer. That is, when exposed for some time to some appropriate kind of physical environment, the bearer will participate in activities that realize the disposition.

primitive? true

why primitive Making the notion of "appropriate physical environment" fully precise is beyond the scope of IDO.

example The disposition of a steel bar to corrode, the disposition of a gasket to leak.

comment An object may not lose a disposition without undergoing physical change.

comment Inspired by the BFO class ‘disposition’ (BFO_0000016).

see also obo:BFO_0000016

comment BFO_0000016 is ‘disposition’.

see also lis:hasDisposition

5.4.2.3 lis:Capability

definition A ‘disposition’, the realization of which serves a positive interest.

primitive? true

why primitive Making the notion of “positive interest” fully precise is beyond the scope of IDO.

example The capability of a pump to serve as an anchor (ISO 15926-2, section 5.2.24.2).

see also https://philarchive.org/rec/MERC-14

comment URI of the paper Merrell et al., 2022, “Capabilities”.

comment In Merrell et al., 2022, ‘capability’ is defined in terms of “interest in”, specifically as a “disposition in whose realization someone (or some organism, or some group of organisms) has or had an interest.”

see also lis:hasCapability

comment The term ‘capability’ appears as a class in 6 different OBO Foundry ontologies, with identifiers obo:SIO_000112, obo:MF_0000043, and obo:NCIT_C74979.

see also obo:SIO_000112

comment SIO_000112 is ‘capability’.

see also obo:MF_0000043

comment MF_0000043 is ‘capability’.

see also obo:NCIT_C74979

comment NCIT_C74979 is ‘capability’.

5.4.2.4 lis:Function

definition A ‘capability’ which is the reason for existence of the bearer. The bearer’s physical design (or evolved form, in the case of natural biological entities) is there in order to enable participation in activities of the specified kind, in a specified way.

primitive? true

why primitive Making the notion of purpose, or “existence for being”, fully precise is beyond the scope of IDO.

example The capability of a motor to apply torque to a shaft.

usage note The typical ‘function’ of a hammer is realized when the hammer participates as the tool in a successful nail-driving activity. If the hammer has multiple functions, such as extracting nails using a claw or shaping objects with a ball, each function should be represented with a distinct individual.

comment Inspired by the BFO class “function” (BFO_0000034).

see also lis:hasFunction

see also obo:BFO_0000034

comment BFO_0000034 is ‘function’.

related in ISO 15926 lis2:FunctionalPhysicalObject

related in ISO 15926 lis12:FunctionalPhysicalObject

5.4.2.5 lis:Role

definition A potential for participation in activities that is assigned to a bearer, and not inherent to the the bearer's physical make-up.

primitive? true

why primitive Making the notion of assignment (to a task) fully precise is beyond the scope of IDO. With complex intentions notions of obligation, which are often required in role assignments, this is particularly clear.

example The role of an interior space (a 'site') as a meeting room is realized when it serves as the space for conducting meetings. The site may be assigned a storage room role at a different time.

comment As with any 'potential', the existence of a role is dependent on its bearer. A particular meeting room's role may therefore not be transferred to a different room; a different role individual with similar classification would be needed.

comment This class is intended to serve the same purpose as the ISO 15926-22 'role' entity type, as well as the same-named BFO class (BFO_0000023).

see also lis:hasRole

see also obo:BFO_0000023

comment BFO_0000023 is 'role'.

related in ISO 15926 lis2:Role

related in ISO 15926 lis2:RoleAndDomain

EXAMPLE. To represent that a manufacturer creates a line of products, or a craftsperson fabricates one-off items, a 'maker role' may be introduced. We stipulate that a 'maker role' is realized only in a 'making' activity which 'has active participant' the role-bearer. The 'activity' class 'making' is introduced as a prerequisite.

```
Class: ex:Making
  SubClassOf: lis:Activity, lis:creates some lis:Artefact

Class: ex:Maker_Role
  SubClassOf: lis:Role, lis:realizedIn only ex:Making
```

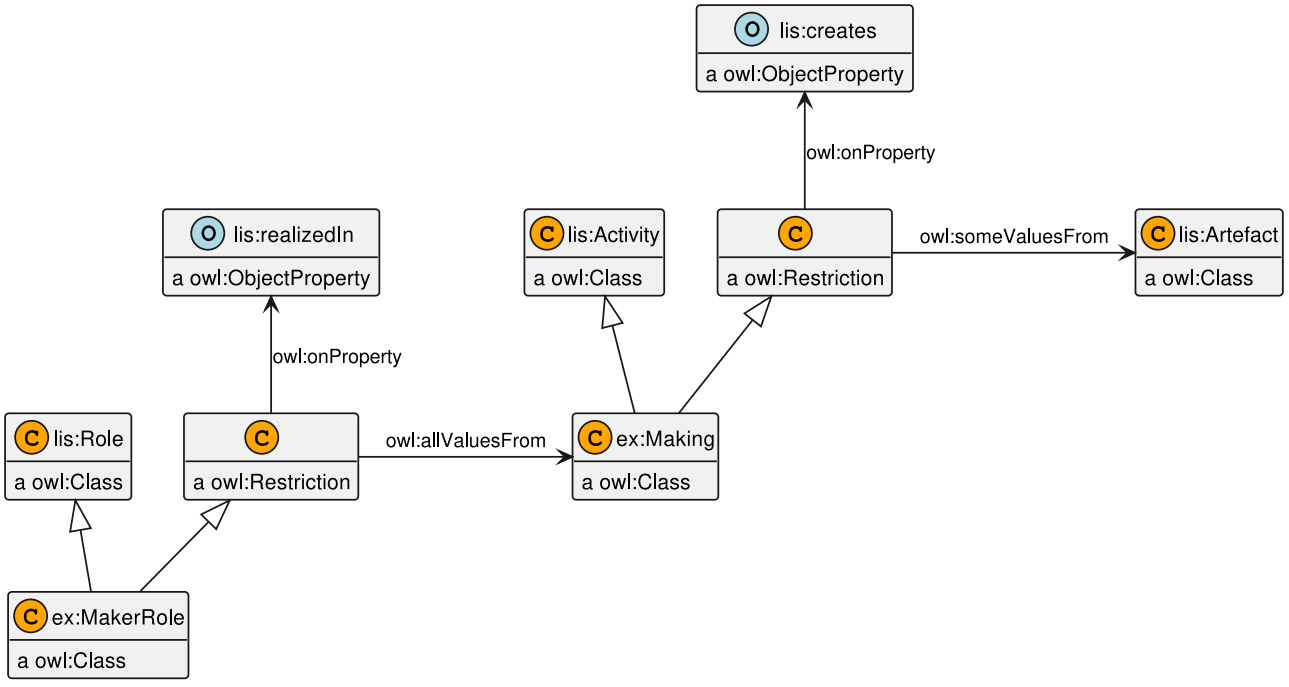


Figure 5.8 — 'role' example, 'maker role'

5.4.2.6 lis:Interest

This class was added on 2023-02-28.

definition A 'potential' which is an attitude of the bearer toward a potential activity.

primitive? true

why primitive Accounting for the notion of (mental) attitude is beyond the scope of IDO.

example A plan to act, an approval of a survey.

5.4.2.7 lis:Quality

definition] A 'dependent [entity' that is present (is actual) any time that the bearer exists.

primitive? true

why primitive Qualities cover a broad range of what is commonly referred to as "properties". It is beyond the scope of IDO to precisely determine what should be included.

example The temperature of a fluid stream, the tensile strength of a bolt, the length of a pipeline, the colour of a surface coating, the familiarity of a specialist with a modification procedure, the quality of a window of being open or closed.

usage note Qualities are in general subject to change over the lifetime of the bearer. The main IDO device for representing quality changes is through 'datum' individuals: see lis:qualityQuantifiedAs.

related in ISO 15926 lis2:Property

explanatory note In ISO 15926-2, the term "quality or characteristic" is used to explain what the entity type 'property' represents (section 5.2.26.5, notes 1 and 2). Note 2 reads, "The types of characteristic or quality, such as temperature or density, are instances of class_of_property."

related in ISO 15926 lis12:Quantity

comment The class Quality and its subclass PhysicalQuantity are inspired by corresponding classes included in the DOLCE and BFO (BFO_0000019) upper ontologies.

see also obo:BFO_0000019

comment BFO_0000019 is 'quality'.

see also obo:PATO_0000001

comment PATO_0000001 is 'quality'.

see also obo:PATO_0001241

comment PATO_0001241 is 'physical object quality'. In the Information Artifact Ontology, defined as "A quality which inheres in a continuant".

see also dol:quality

comment The DOLCE definition of 'quality': "Qualities can be seen as the basic entities we can perceive or measure: shapes, colors, sizes, sounds, smells, as well as weights, lengths, electrical charges… 'Quality' is often used as a synonymous of 'property', but this is not the case in this upper ontology: qualities are particulars, properties are universals. Qualities inhere to entities: every entity (including qualities themselves) comes with certain qualities, which exist as long as the entity exists."

see also ssn:Property

comment The SSN definition of 'Property' (2017): "A quality of an entity. An aspect of an entity that is intrinsic to and cannot exist without the entity."

see also ssn:ActuatableProperty

comment The SSN offers the following as an example for ActuatableProperty, a subclass of Property: "A window actuator acts by changing the state between a frame and a window.

The ability of the window to be opened and closed is its ActuatableProperty." Note that IDO doesn't provide an analogue.

5.4.2.8 lis:PhysicalQuantity

definition A 'quality' that can be quantified according to a unit of measure.

primitive? true

why primitive The notion of "unit of measure" is employed in the definition of 'physical quantity'. It is beyond the scope of IDO to give an account of measurement.

example The mass of a bar of steel, the temperature of a medium, the inner diameter of a pipe, the lift height of a crane, the head (lift height) of a pump.

related in ISO 15926 lis2:Property

equivalent in ISO 15926 lis12:PhysicalQuantity

comment ISO 15926-12 defines PhysicalQuantity as, "<Quantity> that is measurable or observable".

see also obo:PATO_0001018

comment PATO_0001018 is 'physical quality'. In the Information Artifact Ontology, defined as "A quality of a physical entity that exists through action of continuants at the physical level of organisation in relation to other entities." Note that this is distinct from 'physical object quality'; see annotation to lis:Quality.

see also dol:physical-quality

comment The DOLCE definition of 'physical-quality': "A quality inherent in a physical endurant."

see also ssn:ObservableProperty

comment The SSN offers the following as an example for ObservableProperty, a subclass of Property: "The height of a tree, the depth of a water body, or the temperature of a surface are examples of observable properties, while the value of a classic car is not (directly) observable but asserted."

see also http://qudt.org/schema/qudt/QuantityKind

comment QUDT explains 'quantity kind' as, "any observable property that can be measured and quantified numerically. Familiar examples include physical properties such as length, mass, time, force, energy, power, electric charge, etc. Less familiar examples include currency, interest rate, price to earning ratio, and information capacity."

see also om:Quantity

comment OM2 explains 'quantity' as, "A quantity is a representation of a quantifiable (standardised) aspect (such as length, mass, and time) of a phenomenon (e.g., a star, a molecule, or a food product). Quantities are classified according to similarity in their (implicit) metrological aspect, e.g. the length of my table and the length of my chair are both classified as length."

5.4.2.9 lis:ShapeQuality

NOTE. Tentative addition *[2023-02-15]*.

definition A 'quality' that can be characterised in terms of spatial properties.

primitive? true

why primitive A precise account of of "spatial properties" is beyond the scope of IDO.

example The flat shape of a surface feature, the cylindrical shape of a piece of pipe.

5.4.2.10 lis:MaterialCompositionQuality

NOTE. Tentative addition *[2023-02-15]*.

definition A 'quality' that can be characterised in terms of chemical properties.

primitive? true

why primitive A precise account of of "chemical properties" is beyond the scope of IDO.

example The steel grade of a cast-iron component, the plastic type of a wire sheathing.

5.4.2.11 lis:PhaseQuality

NOTE. Tentative addition *[2023-02-15]*.

definition A 'quality' that can be characterised in terms of material phase.

primitive? true

why primitive A precise account of of "material phase" is beyond the scope of IDO.

example The liquid quality of an amount of molten steel, the solid quality of a steel gasket.

deprecated? true

comment This class is included for coverage of ISO 15926-2, which has 'phase', but many classes don't have a need for it.

5.4.3 lis:Temporal

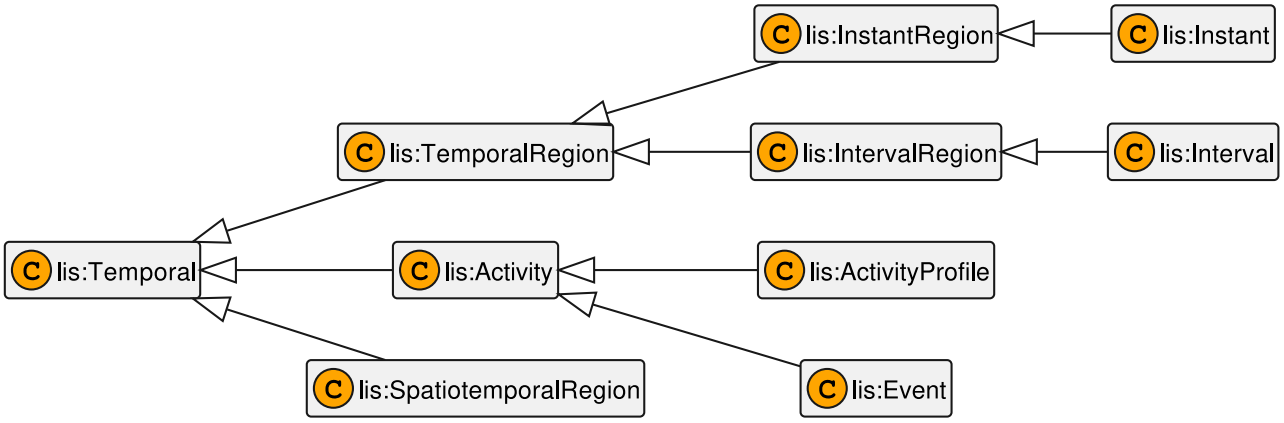


Figure 5.9 — IDO 'temporal' class hierarchy

alternative label Temporal Entity

alternative label Occurrent

comment "Temporal" is a candidate better name, 2023-03-20.

definition A part of the flow of time, or something that takes place in time.

primitive? true

why primitive A comprehensive account of time, and of the unfolding of events in time, is beyond the scope of IDO.

example A moment in time, a period of time, an activity or process, a beginning, a closing event.

comment This class is tentatively introduced Q1 2023 as a supertype of any temporal (including 4D) entities. Part 14 has, until 2023, had Activity in the same ultimate role as 'temporal [entity]'. However, as in ISO 15926-2 (see annotations and the text of the standard), we require participants for any Activity, but it's not obvious that one should assume participants for points or periods in time. This motivates introducing a more generic superclass like 'temporal'.

comment On the relationship between processes in BFO and DOLCE, see the presentation ProcessOntology and the paper Classifying Processes and Basic Formal Ontology (Jarrar and Ceusters, 2017).

comment In ISO 15926-2, Activity is not a supertype of Event; in BFO, 'process' is not a supertype of 'process boundary'. This may indicate that 'event' in IDO should in fact not be a subclass of 'activity'.

see also obo:BFO_0000003

comment BFO_0000003 is 'occurrent'.

see also dol:perdurant

comment From the DOLCE description of 'perdurant': "Perdurants (AKA occurrences) comprise what are variously called events, processes, phenomena, activities and states. …"

```
DisjointClasses: lis:Activity, lis:SpatiotemporalRegion, lis:TemporalRegion
```

5.4.3.1 lis:Activity

definition A 'temporal' entity that 'has participant' some 'physical object'.

primitive? true

why primitive (Here we could consider saying the class is in fact not primitive, but defined in terms of 'temporal' and 'has participant'.)

example A pumping, an inspection, a report writing, a fastening (the activity of a bolt in holding two pieces together).

scope note An 'activity' doesn't need to be temporally connected. For example, intermittent operation of a heater over a week can be represented by a single 'activity' individual; it will occupy a non-connected 'interval region'. The details of each connected activity part ("singular activities") may be modelled by relating the activity to 'interval' individuals.

equivalent in ISO 15926 lis2:Activity

equivalent in ISO 15926 lis12:Activity

see also obo:BFO_0000015

comment From Part 2, section 4.7.17: "… activities are considered to be space-time extensions in which other individuals and events participate."

comment In Part 2 (as well as in BFO, for 'process', intuitively corresponding to Activity), the informal documentation/elucidation states that any activity has a participant. This could be added as an OWL constraint.

comment In BFO-ISO, a constraint (as given in the elucidation, not in an OWL axiom) on 'process' is that it has "some temporal proper part". This is not included in IDO. On the contrary, we make 'event', which will have an instant as temporal extent, a subclass of 'activity'.

see also dol:accomplishment

see also dol:process

5.4.3.2 lis:ActivityProfile

definition A part of an 'activity', a "selective abstraction" (cf. BFO reference) to just that part of the containing activity that consists in varying or maintaining a specified 'quality' of a participant.

primitive? true

why primitive A precise account of the notion of selective abstraction is beyond the scope of IDO.

example The BFO 2.0 reference provides good examples, like "constant speed profile", "acceleration profile", and "irregular beat profile".

usage note Typically, a physical quantity of a participant, such as a "pressure" individual, is selected, and the "pressure profile" of the activity is introduced. The relation 'profile of quality' links to that pressure quality individual that defines (is the basis for abstraction to) the activity profile. By means of the relation 'profile quantity datum', the activity profile can be related to 'quantity datum' individuals, to which aggregation can be applied, giving a basis for classification of the activity profile; for example, as a "rising pressure" activity.

example An 'activity profile' of an 'activity' involving rotating equipment, instantiating a class "100 revolutions per minute".

comment An 'activity profile' is a part of an 'activity', a "selective abstraction" (BFO reference) made according to a quality of interest to the process. Typically, we will select a physical quantity such as pressure or temperature, and refer to the "pressure profile" of the activity. By means of the relation 'profile quantity datum', the activity profile can be related to a set of 'quantity datum' individuals, to which aggregation can be applied, giving a basis for classification.

comment As in BFO (documentation), activities in IDO don't have physical quantities, or, more generally, qualities. However, they may be classified as, for instance, "running at a constant 300 revolutions per minute", with implicit reference to physical quantities. 'activity profile' parts and related relations provide the supporting model.

comment ISO 15926-2 and IDO (and BFO, as mentioned in its documentation) agree that assignment of physical quantities to activities is done through classification. An example of this would be classifying the activity of rotating equipment as "constant 300 rpm." However, they diverge on physical entities. ISO 15926-2, which recognises only 4D entities, mandates the classification approach for any physical entity (see, e.g., the example provided for section 5.2.9.9, p. 129: "… the room was a member of the 20 Celsius property"). IDO (and BFO) represents such assignment using quality individuals dependent on 3D objects; cf. lis:hasPhysicalQuantity and related resources.

comment The name "ActivityProfile" is perhaps unfortunate since 'process profile' may be considered an established term. It has been chosen, tentatively, because lis:Activity corresponds to BFO's 'process'. Other candidate names are "process profile", as in BFO, and "activity/process projection", where the latter may be said to better reflect the intended meaning.

comment BFO 2.0 reference, section 3.13: "process profile parts […] are of the sort that serve as the target of a process of measurement. The key to annotating many process measurement data in BFO terms is to identify the process profiles represented by the corresponding measurement charts created in the salient domains."

comment The BFO class BFO_0000144 'process profile' is described in the BFO 2.0 reference specification. It is however not included in BFO-ISO or the OBO Relation Ontology. The class appears in more than 50 different OBO ontologies, according to the Ontobee search facility; note that the supporting relation 'process profile of' appears in only 11 of these.

comment From BFO, see "Classifying processes", section 4.2 (Smith, 2012); BFO 2.0 Reference, section 3.13 (Smith, 2015). On relational qualities, like for "delta" pressure, see BFO book, p. 97 f. (Arp et al. 2015).

adapted from obo:BFO_0000144

comment BFO_0000144 is 'process profile'.

EXAMPLE. A car drives from A to B at an average speed of 50 km/h; categorise the activity, and relate the average speed to observed values. We will incrementally refine the level of detail in our model for this fact, to illustrate the usage of 'activity profile'.

Introduce *mydrive* as a 'travel' activity, and introduce the ad-hoc object property 'travel speed' to relate the activity to an 'average datum'.

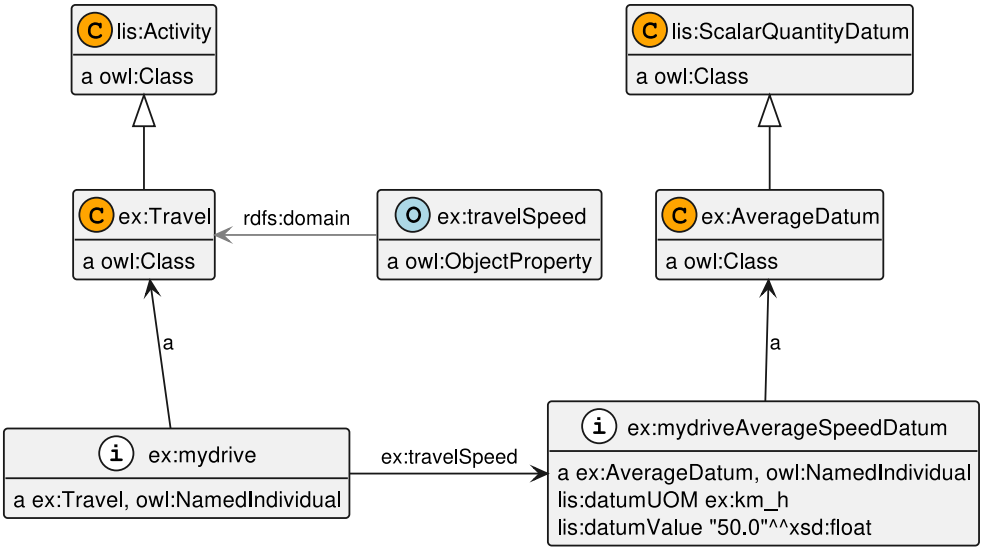


Figure 5.10 — 'activity profile' example: relating activity to aggregate datum

To have OWL reasoning automatically classify *mydrive* as belonging to a category of 'travel at average 50 km/h', a class with OWL equivalence restrictions as follows may be added to the ontology. (A class definition diagram is omitted here because the RDF representation is more verbose and less readable than the OWL axioms.)

```
Class: ex:TravelAverage50km_h
  EquivalentTo: ex:travelSpeed some
    (ex:AverageDatum
      and (lis:datumUOM value ex:km_h)
      and (lis:datumValue some xsd:float[>= 45.0f , < 55.0f]))
```

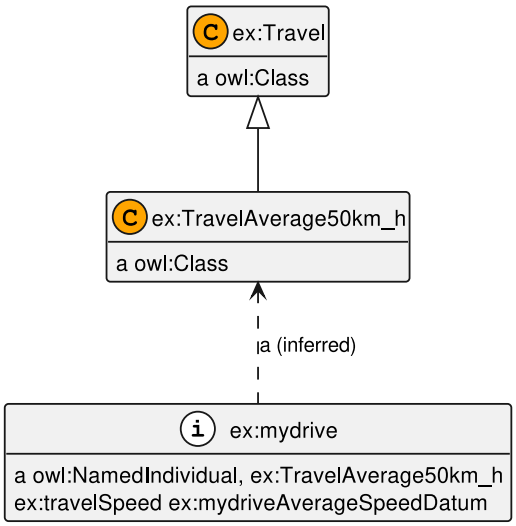


Figure 5.11 — 'activity profile' example: inference to category by speed range

Introduce *mycar* as an 'active participant' in the activity, with 'physical quantity' *mycar_speed*, the quality that is quantified by the average-speed datum.

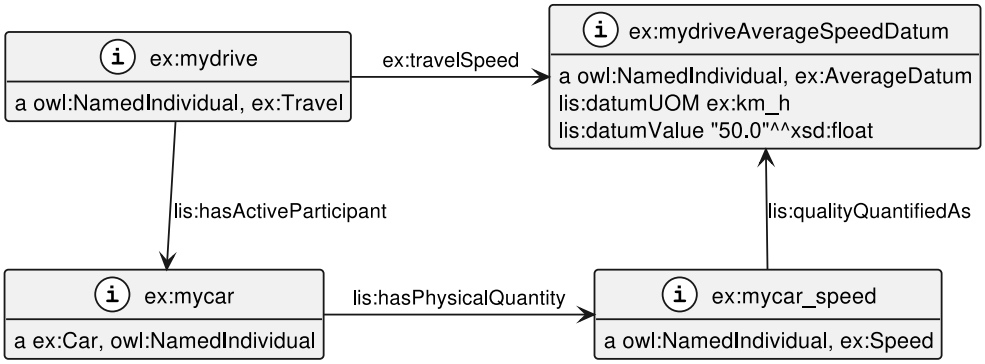


Figure 5.12 — 'activity profile' example: participant with quantity

Provide the average-speed datum with a basis in observations over time. For this, the graph needs to provide two paths from the driving activity to each observation:

Via the participant, to represent that the observation datum describes the speed quality of the car.

Via an activity profile, a *selective abstraction* (projection) from the driving activity based on the speed quality of the car, to represent that the datum belongs to the activity.

Thus, introduce an 'activity profile' for *mycar_speed*, and relate it to observed 'speed datum' individuals. The diagram shows one such datum 'my drive speed datum 1', with timestamp. The dotted lines in this diagram correspond to relationships that will be inferred by IDO property paths (on the relations 'profile of quality' and 'has participant'), helping ensure consistency of the model.

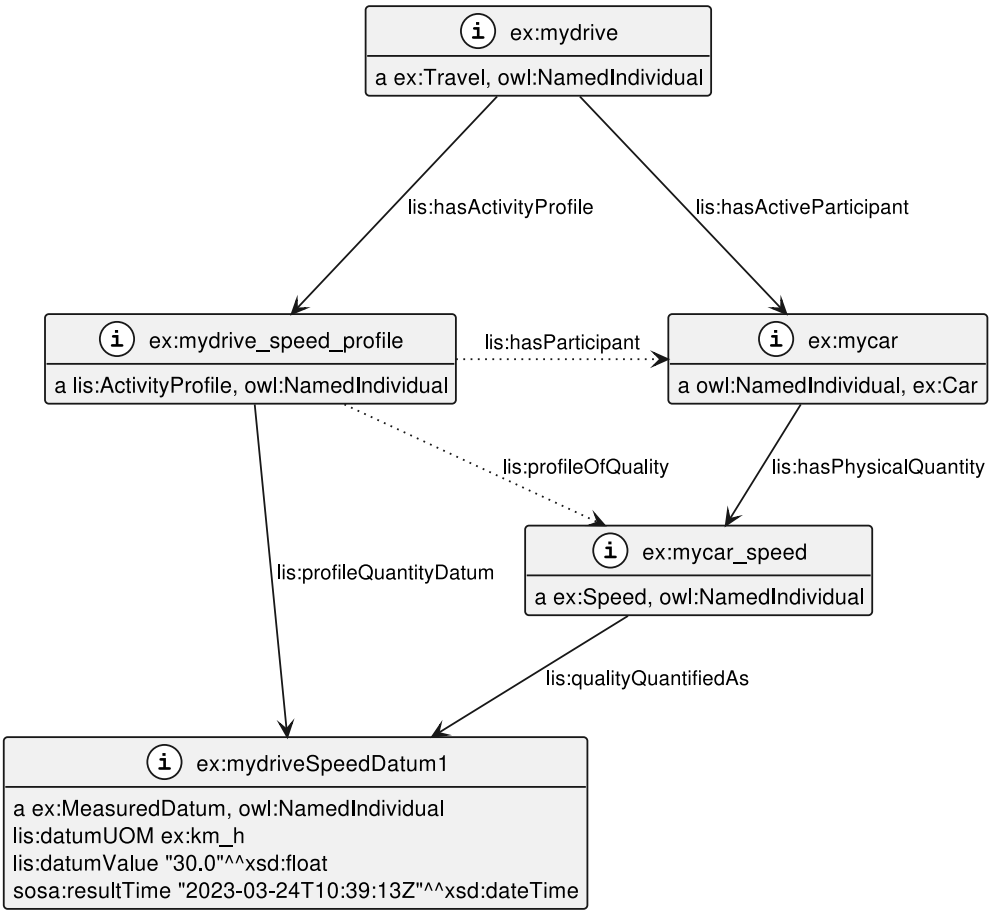


Figure 5.13 — 'activity profile' example: profile and participant data

Introduce a container for the "data set" of measured datum individuals, and relate it to the average-speed datum by an 'averaged to' relation.

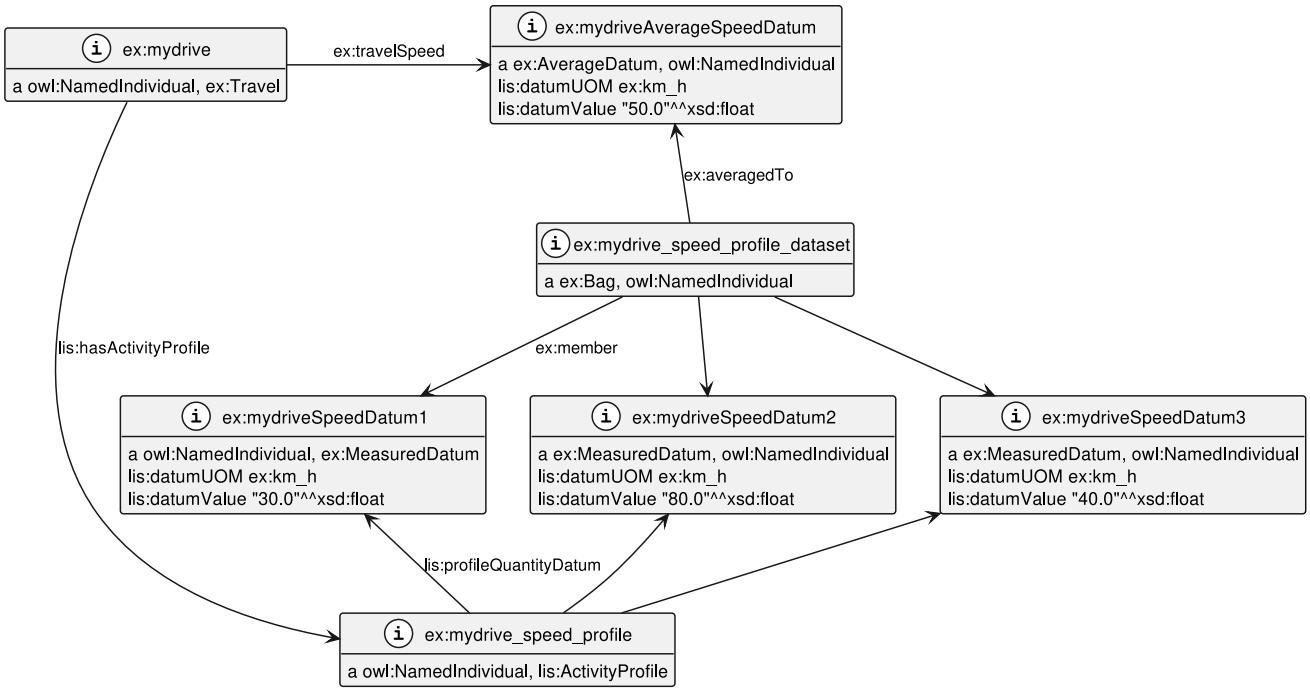


Figure 5.14 — 'activity profile' example: aggregation from observed data

Note 1. The RDF vocabulary for containers is not available for use in an OWL ontology. Resources like ex:Bag and ex:member may however be introduced, here mirroring rdf:Bag and rdfs:member.

Note 2. Mathematical calculations are outside the scope of OWL reasoning, so the 'averaged to' relationship would need to be derived by external means (in a simple case like this, a SPARQL query would suffice). The model is however suitable for representing the result.

Note 3. In the figure, each datum is represented by an OWL individual. In a use case involving large sets of measurements, this would be inefficient. To avoid inefficiency, one option would be to provide a pointer from the dataset individual to a data source kept externally to the semantic asset model.

Figure 5.1 shows the combined graph of individuals described above.

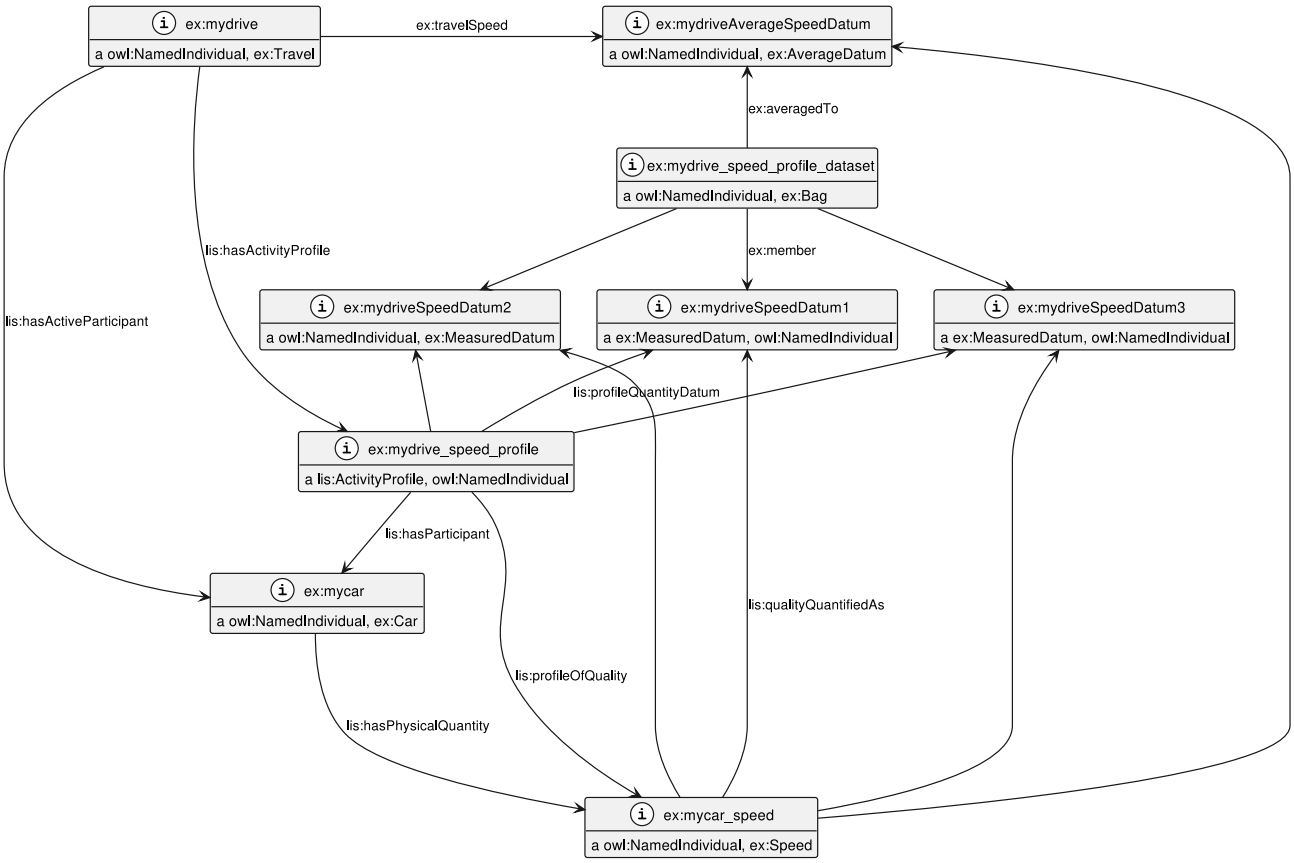


Figure 5.15 — 'activity profile' example: full diagram of individuals

5.4.3.3 lis:Event

definition An instantaneous part of an 'activity', i.e., its temporal extent is an 'instant'.

primitive? true

why primitive (Here we could consider saying the class is in fact not primitive, but defined in terms of 'activity', 'has temporal extent', and 'instant'.)

example The beginning of a lifting operation, the end of a measurement activity, the midpoint between beginning and end of an activity.

comment An 'event' doesn't need to be temporally connected. For example, several instances of switching on a heater over a week may be represented by a single 'event' individual; it will occupy a non-singular 'instant region'. The details of each singular event part may be modelled by relating the event to 'instant' individuals.

equivalent in ISO 15926 lis2:Event

equivalent in ISO 15926 lis12:Event

see also obo:BFO_0000035

comment BFO_0000035 is 'process boundary'.

comment Event is close in meaning to BFO's 'process boundary', and to DOLCE's 'achievement'. Note that BFO's 'process boundary' is a sibling class to 'process', even though the definition annotation states it to be a "temporal part of a process". It's not obvious why the event/process boundary shouldn't have the same participants as the parent activity/process.

see also dol:achievement

5.4.3.4 lis:SpatiotemporalRegion

definition A 4D region, extended in space as well as time.

primitive? true

why primitive A precise account of the notion of spacetime goes beyond the scope of IDO.

example The spatiotemporal region occupied by the fluid contents of a vessel over a period of time, the spatiotemporal region occupied by a pipeline inspection device (pig) during an inspection.

comment An 'activity' occupies a 'spatiotemporal region'. Note that the spatiotemporal region doesn't need to be connected.

see also obo:BFO_0000011

comment BFO_0000011 is 'spatiotemporal region'.

see also dol:spatio-temporal-region

5.4.3.5 lis:TemporalRegion

definition An extent of time.

primitive? true

why primitive A precise account of the notion of time goes beyond the scope of IDO.

example The moment a valve is closed, upstart time for a motor, the length of time that a bolt has been installed, intervals between maintenance.

see also obo:BFO_0000008

comment BFO_0000008 is 'temporal region'

see also dol:temporal-region

5.4.3.6 lis:InstantRegion

definition An individual that has one or more separated (cf. BFO-ISO 'zero-dimensional temporal region') instants as parts.

primitive? true

why primitive (Could be considered non-primitive, defined in terms of 'temporal region' and each part mapping by 'has temporal extent' to 'instant'.)

example The moment a valve is closed, the instants of a series of measurements.

5.4.3.7 lis:Instant

definition A single point in time.

primitive? true

why primitive (Could be considered non-primitive, defined in terms of 'instant region' and 'no proper part'.)

example The moment a valve is closed, the time of a measurement as use for time-stamping, the moment an activity starts.

synonym PointInTime

comment The ISO 15926-2 term is 'point in time'. This has been deprecated in IDO for more consistent naming, within IDO and with the OWL Time ontology.

equivalent in ISO 15926 lis2:PointInTime

equivalent in ISO 15926 lis12:PointInTime

see also time:Instant

comment 'instant' is defined in OWL Time as "A temporal entity with zero extent or duration".

5.4.3.8 lis:IntervalRegion

definition An individual that has one or more separated intervals as parts.

primitive? true

why primitive (This class could be considered non-primitive, defined in terms of 'temporal region' and each part mapping by 'has temporal extent' to 'interval'.)

example Intermittent times of operation of a heater, maintenance periods at a plant, times that a milling machine is run, The Saturdays of 2011, the construction period for a plant modification

see also obo:BFO_0000038

comment BFO_0000038 is 'one-dimensional temporal region' in BFO-ISO.

5.4.3.9 lis:Interval

definition A continuous period of time.

primitive? true

why primitive A precise account of temporal intervals is beyond the scope of IDO (temporal logic offers no obvious consensus).

example The time an instrument has been in operation, the time during which a measurement was made.

synonym PeriodInTime

comment The ISO 15926-2 term is 'period in time'. This has been deprecated in IDO for more consistent naming, within IDO and with the OWL Time ontology.

equivalent in ISO 15926 lis2:PeriodInTime

equivalent in ISO 15926 lis12:PeriodInTime

see also obo:BFO_0000202

comment BFO_0000202 is 'temporal interval'.

see also time:Interval

comment 'interval' is defined in OWL Time as "A temporal entity with an extent or duration".

see also dol:time-interval

5.4.4 lis:Prescriptive

NOTE. Tentative addition *[2023-03-01]*.

definition An entity that is part of a specification.

primitive? true

why primitive The notion of specification is not defined. An explication should be given in terms of systems engineering practices.

example A "tag" individual, a function of a tag, an activity as characterised in process design.

usage note This class is not intended to be used directly, but only for serving as the domain of 'installed as' and similar relations. The disjointness axiom serves to block iterated chains of 'installed as', etc., relationships.

comment] A 'prescriptive [entity' is an individual that represents the asset as designed, with restrictions that express requirements.

comment A 'prescriptive' individual may be matched with an individual representing an 'actual' individual. Naming for the according relations is tentative: 'installed as' for objects,; 'executed as' for activities; 'implemented by' for dependent entities.

see also lis:Actual

```
DisjointClasses: lis:Prescriptive, lis:Actual
```


5.4.4.1 lis:PrescriptiveObject

NOTE. Tentative addition *[2023-02-27 Mon]*.

definition An 'object' that is part of a specification.

primitive? true

why primitive An account of the nature of specifications is beyond the scope of IDO.

example An equipment tag, a functional location (SAP), a maintenance schedule, a production area, a software program for instrument control, a CAD design.

see also lis:InstalledObject

see also lis:installedAs

explanatory note The class 'prescriptive object' may be considered defined as "an individual that may appear on the right side of an 'installed as' relationship". It is in this sense not primitive.

```
Class: lis:PrescriptiveObject
  SubClassOf: lis:Object
```

5.4.5 lis:Actual

NOTE. Tentative addition *[2023-03-01]*.

definition An entity that is a concrete implementation of a 'prescriptive [entity]'.

primitive? true

why primitive The notion of concrete implementation, of providing an actual item to serve a design, is not defined. An explication should be given in terms of systems engineering practices.

example A serial-numbered artefact installed at a tag, a function of a concrete artefact, a run or execution of a designed process, an industrial facility "as built".

5.4.5.1 lis:InstalledObject

NOTE. Tentative addition *[2023-02-27 Mon]*.

[TODO I suggest to change the name to lis:ActualObject to reflect the Objects that are Actual entities, but not installed yet (e.g. spare parts with a serial number MMB 2023-03-23]

definition An actual individual that serves as a prescriptive object for some period of time.

primitive? true

why primitive A full account of the nature of prescription or specification, and actual implementation is beyond the scope of IDO.

example A concrete, serial-numbered artefact that is installed as a "tag" is the paradigm of 'installed object'. See examples for 'prescriptive object'.

see also lis:installedAs

see also lis:PrescriptiveObject

explanatory note The class 'installed object' should be considered defined as "an individual that may appear on the left side of an 'installed as' relationship". It is in this sense not primitive.

```
Class: lis:InstalledObject
  SubClassOf: lis:Object
```

5.5 Individual relations (object properties)

```
## Object property details
```


5.5.1 lis:concretizes

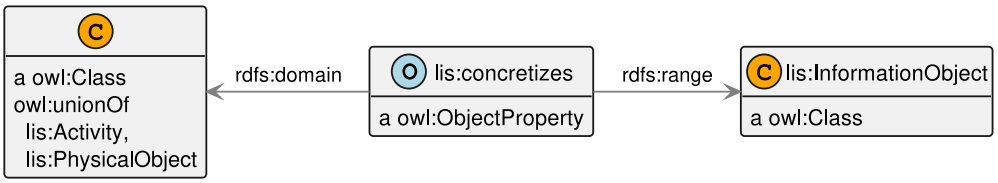


Figure 5.16 — IDO 'concretizes' relation

definition If x 'concretizes' y, then y is an 'information object', and x is a 'physical object' copy of y, or an 'activity' performance of y.

primitive? true

why primitive A comprehensive definition of what it means to be a copy, or manifest a pattern, is beyond the scope of IDO. The definition given here is clearly incomplete.

example A report is produced as paper copies, a spreadsheet document is shown on a computer screen, a plan is carried out, a device operates according to a functional specification, a software program is executed.

see also lis:concretizedBy

usage note There are two main, and very different, uses of 'concretizes' for an information object y. (1) The information object is made concrete in printed or electronic copies. Each copy will have patterns of ink or configurations of electrons that map to y: a quality of appropriate similarity. (2) The information object is made concrete by a performance according to the specification given by y. A 'process profile' of the performance will map to the pattern of y in an appropriate way.

usage note The relation of concretization is under-documented, especially when it comes to concretization as entities to be realized in activities, or activities themselves; we follow the latter, as is done in BFO-ISO, where the performance of a musical piece is the concretization. Other BFO documentation says the intention to perform is the concretization. The current annotations and axioms in IDO represent a best effort, but may need to be changed.

scope note Inspired by BFO's "concretizes". The constraints on 'concretizes' in IDO closely matches that employed in BFO-ISO (obo:BFO_0000059), and not the RO or the IAO (obo:RO_0000059).

see also obo:RO_0000059

comment RO_0000059 is 'concretizes', from the Relations Ontology; also used in the Information Artifact Ontology.

see also obo:BFO_0000059

comment BFO_0000059 is 'concretizes at some time', from BFO-ISO

comment BFO-ISO has 'process or specifically dependent continuant' as the domain of the concretization relation. This is problematic, as a 'process' is not a realizable entity, which fails to match earlier BFO documentation: see Building Ontologies with BFO, p. 107; and BFO reference, p. 66, where the concretization of a 'plan' is clearly stated to be a 'realizable entity' (to be realized in a process), in "You may concretize a recipe that you find in a cookbook by turning it into a plan which exists as a realizable dependent continuant in your head.". However, considering the recently improved 'process profile' materials, we follow BFO-ISO.

see also obo:BFO_0000164

comment BFO_0000164 is 'concretizes at all times', from BFO-ISO

related in ISO 15926 lis12:InformationObject

```
ObjectProperty: lis:concretizes
  Domain: lis:PhysicalObject or lis:Activity
  Range: lis:InformationObject
```

EXAMPLE. A *PDF Document* exists both as a file on a storage medium, and as a printed copy on a bookshelf.

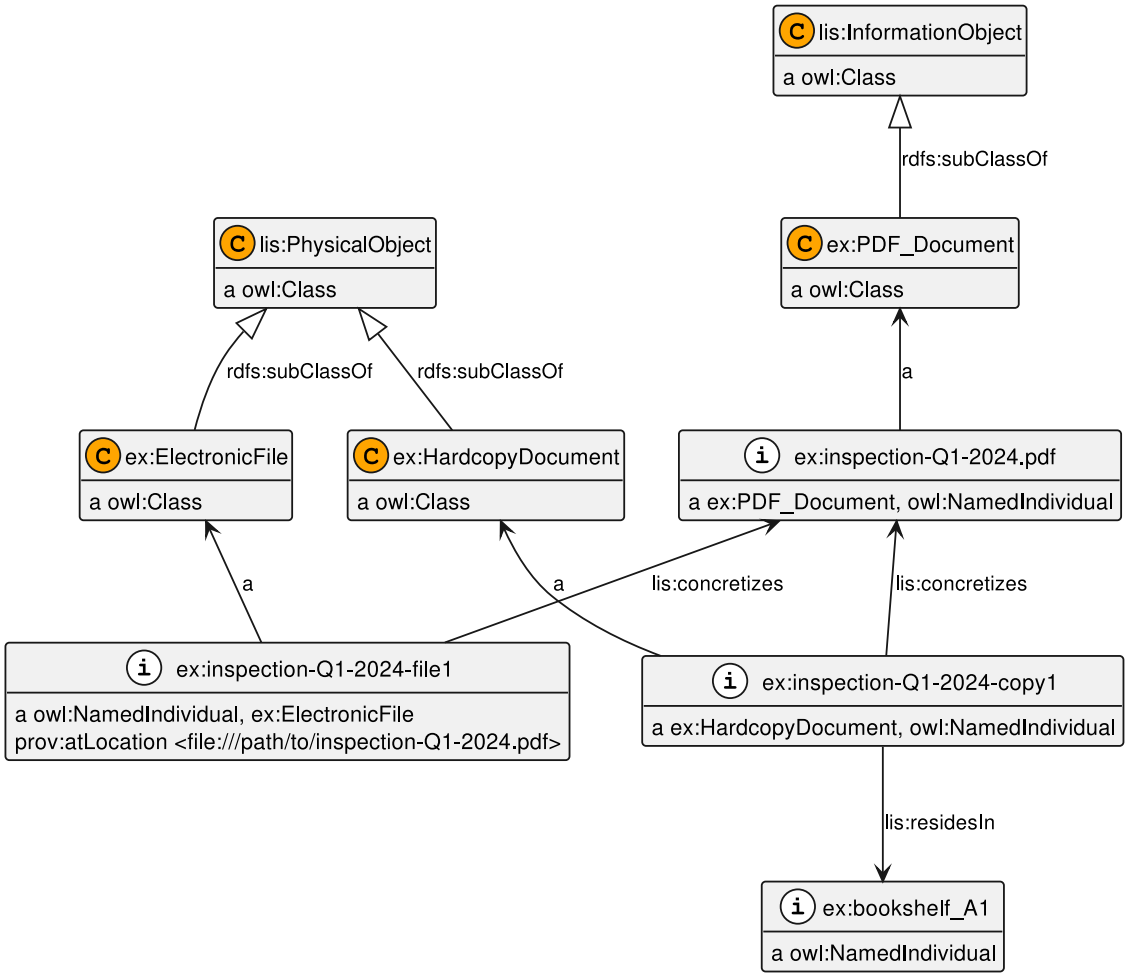


Figure 5.17 — 'concretizes' example

5.5.2 lis:concretizedBy

see also lis:concretizes

usage note For annotations, see the inverse relation 'concretizes'.

primitive? false

```
ObjectProperty: lis:concretizedBy
  InverseOf: lis:concretizes
```

5.5.3 lis:connectedTo

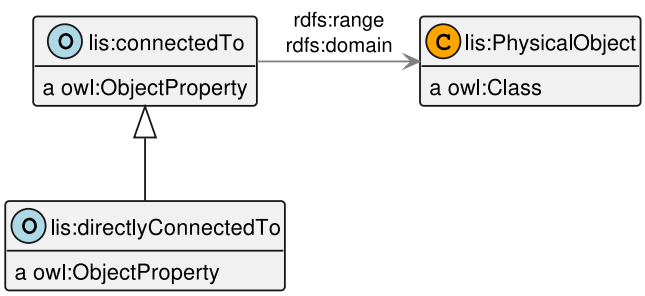


Figure 5.18 — IDO 'connectedTo' relations hierarchy

definition If x is ‘connected to’ y, then there is a physical connection between x and y.

primitive? true

why primitive The notion of being connected to is primitive; an account would require an account of connection activities and constancy of material objects between the connected individuals.

example A three-pin electrical plug is connected to a three-pin socket, a drip pipe is indirectly connected to a drain funnel (examples from ISO 15926-2), two flanges are connected with bolts and a gasket.

see also obo:RO_0002170

comment RO_0002170 is ‘connected to’, in the Relations Ontology.

equivalent in ISO 15926 lis2:ConnectionOfIndividual

equivalent in ISO 15926 lis12:connectedTo

```
ObjectProperty: lis:connectedTo
  Domain: lis:PhysicalObject
  Range: lis:PhysicalObject
  Characteristics: Symmetric
```

5.5.3.1 lis:directlyConnectedTo

definition If x is ‘directly connected to’ y, then x is ‘connected to’ y, and any material present between x and y directly serves to maintain the connection.

primitive? true

why primitive The notion of maintaining connection is not defined; what counts as a direct connection is hard to define precisely.

example An electrical plug is connected to a socket, a tag plate is welded to a pressure vessel.

equivalent in ISO 15926 lis2:DirectConnection

equivalent in ISO 15926 lis12:directlyConnectedTo

```
ObjectProperty: lis:directlyConnectedTo
  Characteristics: Symmetric
```

5.5.4 lis:siteOf

Figure 5.19 — IDO ‘site of’ relation

definition If x ‘site of’ y, then x is a ‘site’, y is an ‘activity’, and y takes place entirely within x.

primitive? true

why primitive x ‘site of’ y means, the spatial extent of the site contains the spatial extent of the activity for the duration of the activity. This requires a relation of “spatial projection throughout the time of the activity”, which is not included in IDO. (The BFO-ISO definition expresses this using the temporalized relation ‘spatially projects onto at all times’.)

example Mixing takes place within a vessel, transport takes place in a pipeline

see also lis:occursIn

synonym environs

see also obo:BFO_0000183

comment BFO_0000183 is ‘environs’ from BFO

see also obo:BFO_0000067

comment BFO_0000067 is ‘contains process’, from RO, where an annotation says the ‘editor preferred term’ is “site of”. There are subtle differences in the range and domain restrictions between BFO’s ‘environs’ and RO’s ‘contains process’: where RO has domain as the broad ‘independent continuant’ only, BFO has (‘site’ or ‘material entity’); where RO has range as ‘occurrent’, BFO has (‘process’ or ‘process boundary’).

see also obo:BFO_0000217

comment BFO_0000217 is ‘spatially projects onto at all times’.

```
ObjectProperty: lis:siteOf
  Domain: lis:Site
  Range: lis:Activity
```

5.5.5 lis:occursIn

see also lis:siteOf

usage note For annotations, see the inverse relation ‘site of’.

primitive? false

see also obo:BFO_0000066

comment BFO_0000066 is ‘occurs in’

```
ObjectProperty: lis:occursIn
  InverseOf: lis:siteOf
```

5.5.6 lis:hasPotential

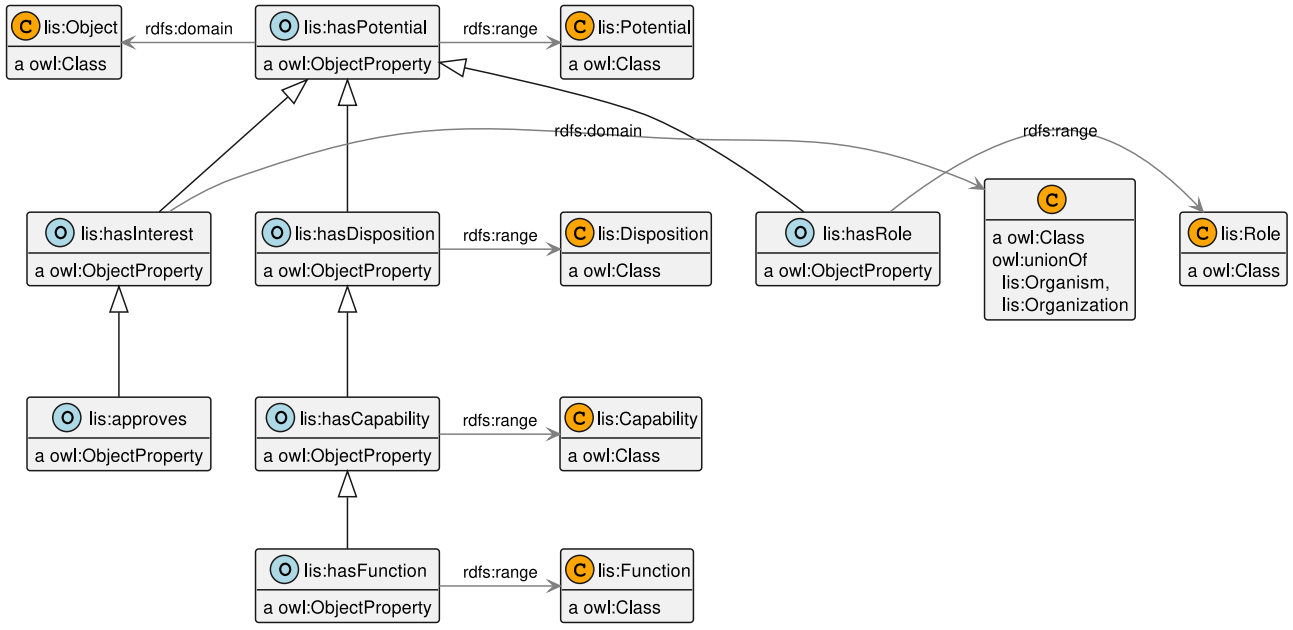


Figure 5.20 — IDO ‘has potential’ relations hierarchy

This relation was added on 2023-02-06.

definition If x ‘has potential’ y, then x is an ‘object’, and y is a ‘potential’ that is dependent on x.

primitive? true

why primitive The relation of dependence is a not defined in IDO.

comment But maybe it should be defined. It’s possible to introduce an analogue of BFO-ISO’s BFO_0000194 ‘specifically depended on by’, as a superrelation to ‘has quality’ and ‘has potential’.

example Having a role, function, obligation, or plan (intention to act).

see also lis:potentialOf

scope note The pre-defined types of 'potential' entities in IDO are 'role', 'disposition', and 'interest'. If other kinds are needed for use case, then 'has potential' is a suitable superrelation for the corresponding bearer relations.

comment Generic relation for assigning 'potential' entities (typically, dispositions or roles) to bearers.

see also obo:RO_0000053

comment RO_0000053 is 'has characteristic', which in the Relation Ontology also has 'quality of' as a subrelation.

see also obo:BFO_0000196

comment BFO_0000196 is 'bearer of', a more generic relation than 'has potential' in that its range is any 'specifically dependent continuant', which in IDO corresponds to 'dependent'.

```
ObjectProperty: lis:hasPotential
  Domain: lis:Object
  Range: lis:Potential
```

5.5.6.1 lis:hasDisposition

definition x 'has disposition' y if and only if x 'has potential' y, and y is a 'disposition'.

primitive? false

example A bolted joint has a disposition to come loose, a motor has a disposition to vibrate, a steel bar has a disposition to rust, a vessel has a disposition to leak, a volatile gas has a disposition to flow upwards.

see also lis:dispositionOf

comment Inspired by BFO's 'has disposition' (RO_0000091). Note that this relation is defined in the Relation Ontology (https://oborel.github.io/) and is not included in BFO-ISO, which provides the more generic 'inheres in' (BFO_0000197).

see also obo:RO_0000091

comment RO_0000091 is 'has disposition'

see also obo:BFO_0000197

comment BFO_0000197 is 'inheres in'

```
ObjectProperty: lis:hasDisposition
  Range: lis:Disposition
```

5.5.6.2 lis:hasCapability

definition x 'has capability' y if and only if x 'has potential' y, and y is a 'capability'.

primitive? false

example A pressure sensor has the capability of being a level sensor, a wall has the capability of noise insulation.

see also lis:capabilityOf

This relation was added on 2023-01-24.

comment Inspired by the introduction of the class 'capability' in recent works by Barry Smith, this relation complements hasDisposition and hasFunction.

comment In the OBO Foundry, we find 'has capability' included only in the Semanticscience Integrated Ontology (SIO), with identifier http://semanticscience.org/resource/SIO_000586. In SIO, 'has capability' is a superrelation to 'has disposition', which is the inverse of what is given in IDO.

```
ObjectProperty: lis:hasCapability
  Range: lis:Capability
```

5.5.6.3 lis:hasFunction

definition x 'has function' y if and only if x 'has potential' y, and y is a 'function'.

primitive? false

example A pump has the function of moving fluids, a door has the function of an explosion barrier.

see also lis:functionOf

comment Inspired by BFO's 'has function' (RO_0000085). Note that this relation is defined in the Relation Ontology (RO, https://oborel.github.io/) and is not included in BFO-ISO, which provides the more generic 'inheres in' (BFO_0000197). Furthermore, in RO 'has function' is a sibling of 'has disposition', not a subrelation.

see also obo:RO_0000085

see also obo:BFO_0000197

related in ISO 15926 lis2:FunctionalPhysicalObject

related in ISO 15926 lis12:FunctionalPhysicalObject

```
ObjectProperty: lis:hasFunction
  Range: lis:Function
```

5.5.6.4 lis:hasRole

definition x 'has role' y if and only if x 'has potential' y, and y is a 'role'.

primitive? false

example An window has the role of a ventilation opening, a room has the role of a storage space, a gas sensor has the role of a leak detector, a person has the role of inspector.

see also lis:roleOf

comment Inspired by BFO's "has role" (RO_0000087)

see also obo:RO_0000087

related in ISO 15926 lis2:Role

```
ObjectProperty: lis:hasRole
  Range: lis:Role
```

5.5.6.5 lis:hasInterest

This relation was updated on 2023-02-28.

definition x 'has interest' y if and only if x 'has potential' y, and y is an 'interest'.

primitive? false

example An intention to start a process, a plan to modify a plant, an approval of a certification.

see also lis:interestOf

comment Note, the use of "potential" in the definition does not imply "not actual", only "not necessarily actual".

explanatory note The relation of interest receives a related, but different treatment in Merrell, et al., "Capabilities" (cf. annotations on 'capability'), p. 14: "We take the 'has an interest in' relation to be primitive, … [T]he domain of this relation is: all organisms and groups of organisms, … [T]he range is process in BFO terms, specifically all processes that are the realizations of dispositions." With IDO, the range is instead a 'potential'; this allows for interests that are not realized.

related in ISO 15926 lis2:LifecycleStage

comment Derived from the relation "LifecycleStage" of ISO 15926-2, which has the attribute pair 'interest'/'interested'; "ClassOfLifecycleStage" provides examples of specialisations as, "EXAMPLE Planned, required, expected, and proposed can be represented by in-

stances of [class_of_lifecycle_stage].” Following this, ‘interest of’ is a superproperty suitable for various intentional relationships, such as planning, approving, or ordering. With IDO, we restrict interest to ‘potential’ entities, while in ISO 15926-2 the domain and range are both unrestricted.

see also obo:IAO_0000033

comment IAO_0000033 is ‘directive information entity’ in the Information Artifact Ontology. See, e.g., subtypes ‘action specification’, ‘objective specification’, ‘plan specification’; and specifically, ‘software’ as a subtype of ‘plan specification’.

```
ObjectProperty: lis:hasInterest
  Domain: lis:Organism or lis:Organization
```

5.5.6.6 lis:approves

deprecated? true

comment The notion of approval will only be employed in a limited range of use-cases, and is therefore too specific to be included in IDO. It may be defined in a domain specific RDL.

comment Relation for stating that some item or activity was approved by an entity, typically a person or an organisation.

equivalent in ISO 15926 lis2:Approval

equivalent in ISO 15926 lis2:Approval

5.5.7 lis:potentialOf

This relation was added on 2023-02-06.

see also lis:hasPotential

usage note For annotations, see the inverse relation ‘has potential’.

primitive? false

comment Generic relation from a ‘potential’ to the bearer (or, for the less common cases of multiply dependent realizable entities, to the bearers).

comment This relation is used in the property chain for ‘has participant’, for inferring participation in a temporal entity (typically, an activity).

```
ObjectProperty: lis:potentialOf
  InverseOf: lis:hasPotential
```

5.5.7.1 lis:dispositionOf

see also lis:hasDisposition

usage note For annotations, see the inverse relation ‘has disposition’.

primitive? false

```
ObjectProperty: lis:dispositionOf
  InverseOf: lis:hasDisposition
```

5.5.7.2 lis:capabilityOf

see also lis:hasCapability

usage note For annotations, see the inverse relation ‘has capability’.

primitive? false

This relation was added on 2023-02-06.

```
ObjectProperty: lis:capabilityOf
  InverseOf: lis:hasCapability
```

5.5.7.3 lis:functionOf

see also lis:hasFunction

usage note For annotations, see the inverse relation 'has function'.

primitive? false

```
ObjectProperty: lis:functionOf
  InverseOf: lis:hasFunction
```

5.5.7.4 lis:roleOf

see also lis:hasRole

usage note For annotations, see the inverse relation 'has role'.

primitive? false

see also obo:RO_0000081

comment RO_0000081 is 'role of' in the Relations Ontology.

related in ISO 15926 lis2:Role

```
ObjectProperty: lis:roleOf
  InverseOf: lis:hasRole
```

5.5.7.5 lis:interestOf

see also lis:hasInterest

usage note For annotations, see the inverse relation 'has interest'.

primitive? false

```
ObjectProperty: lis:interestOf
  InverseOf: lis:hasInterest
```

5.5.7.6 lis:approvedBy

deprecated? true

see also lis:approves

usage note For annotations, see the inverse relation 'approves'.

primitive? false

```
ObjectProperty: lis:approves
  InverseOf: lis:approvedBy
```

5.5.8 lis:installedAs

Figure 5.21 — IDO 'installed as' relation

definition If x 'installed as' y, then x and y are 'object's, and at some time, x is the same individual as y.

primitive? true

why primitive 'has installed' is used to express identity relationships that may vary over time. Because OWL lacks time indices, the representation is informal.

example Pump P-101 is a tag, and ACME pumps with serial numbers 457636 and 457711 have been installed as this pump.

see also lis:hasInstalled

scope note The ‘installed as’ relation is central to the intended use of IDO. It allows for a record of different individuals being installed over time. Typically, a serial-numbered artefact is installed as a ‘prescriptive’ individual (a “tag”) at one time, and later replaced with another serial-numbered artefact. The ‘installed as’ relation carries semantic restrictions only to ensure no object is both installed and prescriptive. No further constraints are given, to allow ‘installed as’ relationships even where restrictions on the prescriptive object are inconsistent with those on the installed object.

usage note Periods of ‘installed as’ relationships should be recorded using OWL annotations, to enable prescriptive objects to be matched with installed objects for selected points or periods in time.

usage note The mechanism for checking requirement consistency on ‘installed as’ relationships must be implemented as a transformation that is external to the OWL ontology.

usage note An asset model with ‘installed as’ relationships supports compatibility checks for installed objects, to uncover possible violations of requirements. This is requires a transformation of the asset model, as follows: replace each ‘installed as’ relationship with identity (owl:sameAs), and replace classifications as ‘prescriptive object’ or ‘installed object’ with just ‘object’ for the participant individuals. Any semantic inconsistency between the the prescriptive object and the installed object may then be discovered by OWL reasoning.

explanatory note The ‘installed as’ relation is declared asymmetric with an OWL restriction. This ensures that or that the relation can’t go both ways, and implies an individual may not be ‘installed as’ itself (asymmetry implies irreflexivity). Furthermore, the domain and range restrictions to disjunct classes ‘installed object’ and ‘prescriptive object’ prevent chains of ‘installed as’ relationships.

```
ObjectProperty: lis:installedAs
  Domain: lis:InstalledObject
  Range: lis:PrescriptiveObject
  Characteristics: Asymmetric
```

5.5.9 lis:hasInstalled

see also lis:installedAs

usage note For annotations, see the inverse relation ‘installed as’.

primitive? false

```
ObjectProperty: lis:hasInstalled
  InverseOf: lis:installedAs
```

5.5.10 lis:realizedIn

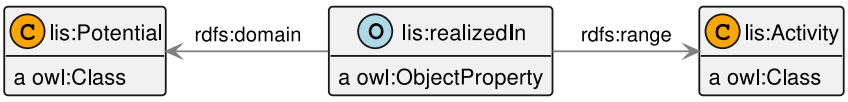


Figure 5.22 — IDO ‘realized in’ relation

definition If x is ‘realized in’ y, then x is a ‘potential’, and y is an ‘activity’ that successfully realizes x.

primitive? true

why primitive A full account of the notion of “successfully realizes” is beyond the scope of IDO. It depends on an account of “success”, in that failed activities will not be realizations, and on the bearer of the potential participating in the realizing activity in the right way.

example A pump realizes its function in a pumping (activity), a bolt realizes its function in a connection, a software monitoring application realizes its function in streaming sensor data.

see also lis:realizes

see also obo:BFO_0000054

comment BFO_0000054 is ‘has realization’ in BFO-ISO, and ‘realized in’ in the Relations Ontology.

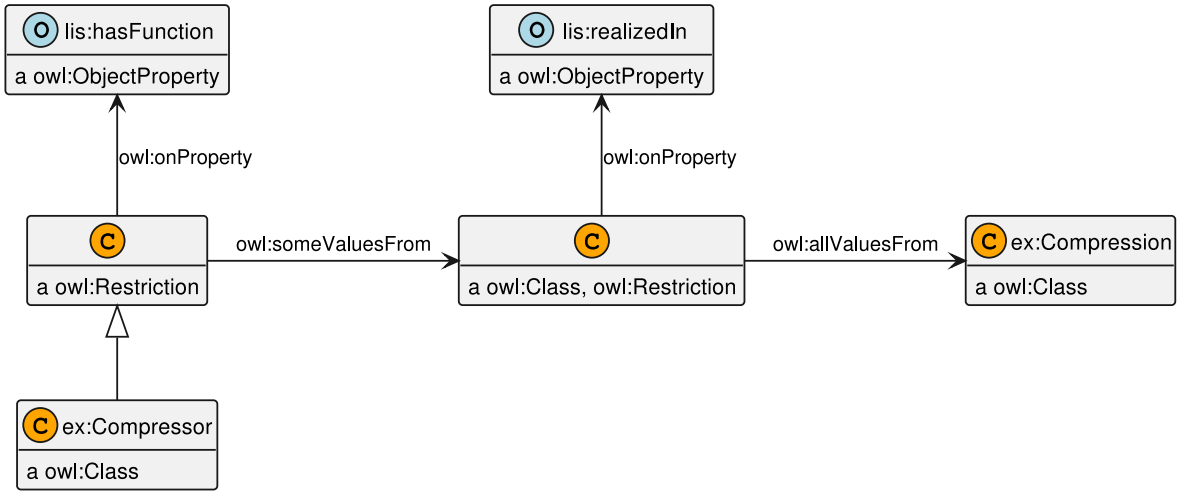
```
ObjectProperty: lis:realizedIn
  Domain: lis:Potential
  Range: lis:Activity
```

EXAMPLE. A *Compressor* has a function that is realized in only *Compression* activities.

OWL axiom expressing the constraint:

```
Class: :Compressor
    SubClassOf: :hasFunction some (:realizedIn only :Compression)
```

RDF diagram corresponding to the OWL axiom:



5.5.11 lis:realizes

see also lis:realizedIn

usage note For annotations, see the inverse relation ‘realized in’.

primitive? false

see also obo:BFO_0000055

comment BFO_0000055 is ‘realizes’ in BFO-ISO.

```
ObjectProperty: lis:realizes
  InverseOf: lis:realizedIn
```

5.5.12 lis:hasQuality

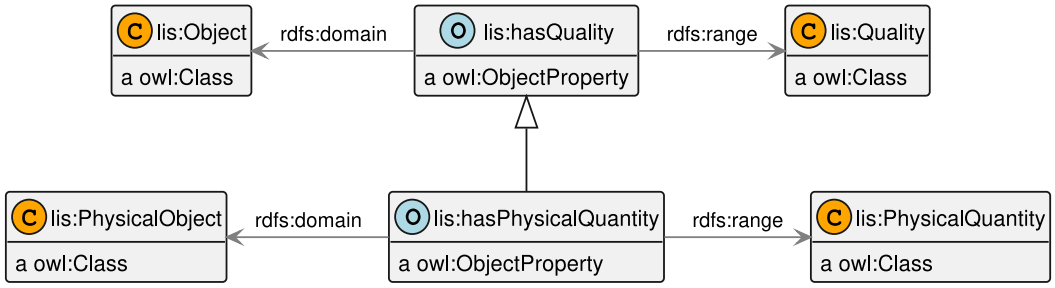


Figure 5.23 — IDO ‘has quality’ relations hierarchy

definition If x ‘has quality’ y, then x is an ‘object’, and y is a ‘quality’ that is dependent on x.

primitive? true

why primitive The relation of dependence is a not defined in IDO.

comment But maybe it should be defined. It's possible to introduce an analogue of BFO-ISO's BFO_0000194 'specifically depended on by', as a superrelation to 'has quality' and 'has potential'.

example A motor has rated output power, a lightbulb has luminance, an amount of liquid has mass, a bar of metal has chemical composition, a document has readability.

see also lis:qualityOf

see also obo:BFO_0000196

comment BFO_0000196 is 'bearer of', a more generic relation than 'has quality' in that its range is any 'specifically dependent continuant', which in IDO corresponds to 'dependent'.

see also obo:RO_0000086

comment RO_0000086 is 'has quality' in the Relations Ontology.

```
ObjectProperty: lis:hasQuality
  Domain: lis:Object
  Range: lis:Quality
```

5.5.12.1 lis:hasPhysicalQuantity

definition x 'has physical quantity' y if and only if x 'has quality' y, and y is a 'physical quantity'.

primitive? false

example A section of pipe has length, a crane has lifting power, a flange has inner diameter.

see also lis:physicalQuantityOf

```
ObjectProperty: lis:hasPhysicalQuantity
  Domain: lis:PhysicalObject
  Range: lis:PhysicalQuantity
```

5.5.13 lis:qualityOf

see also lis:hasQuality

usage note For annotations, see the inverse relation 'has quality'.

primitive? false

```
ObjectProperty: lis:qualityOf
  InverseOf: lis:hasQuality
```

5.5.13.1 lis:physicalQuantityOf

see also lis:hasPhysicalQuantity

usage note For annotations, see the inverse relation 'has physical quantity'.

primitive? false

```
ObjectProperty: lis:physicalQuantityOf
  InverseOf: lis:hasPhysicalQuantity
```

5.5.14 lis:profileOfQuality

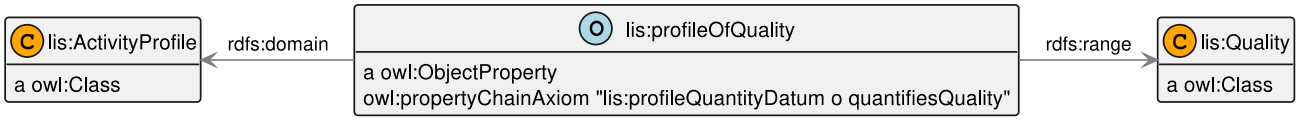


Figure 5.24 — IDO 'profile of quality' relation

definition x is a 'profile of quality' y if and only if x is an 'activity profile' that varies only in the 'quality' y.

primitive? false

example A temperature activity profile of a heating process represents just that part of the process which is a variation in temperature.

see also lis:qualityProfiledIn

scope note 'profile of quality' should be considered a functional property (i.e., an 'activity profile' is a profile of only a single 'quality'). However, in OWL, an object property with property chain restrictions may not be combined with cardinality restrictions. Correct cardinalities must therefore be maintained with rules external to the ontology.

usage note A property chain axiom ensures that a quantity datum individual associated with an activity profile by 'profile quantity datum' quantifies a quality which defines the activity profile by 'profile of quality'.

```
ObjectProperty: lis:profileOfQuality
  Domain: lis:ActivityProfile
  Range: lis:Quality
  SubPropertyChain:
    lis:profileQuantityDatum o lis:quantifiesQuality
```

5.5.15 lis:qualityProfiledIn

This relation was added on 2023-02-24.

see also lis:profileOfQuality

usage note For annotations, see the inverse relation 'profile of quality'.

primitive? false

```
ObjectProperty: lis:qualityProfiledIn
  InverseOf: lis:profileOfQuality
```

5.5.16 lis:isAbout

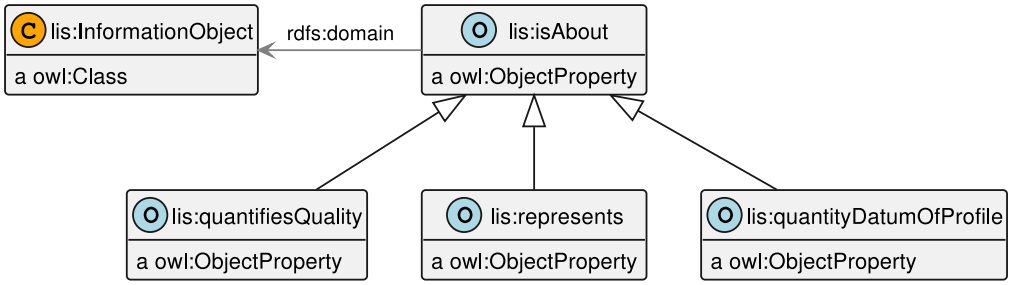


Figure 5.25 — IDO 'is about' relations hierarchy

definition If x 'is about' y, then x is an 'information object' of relevance to y.

primitive? true

why primitive A comprehensive account of aboutness is beyond the scope of IDO.

example A regulation describes a procedure, a datum quantifies a quality, a nameplate provides basic information about a pressure vessel, a log file traces quantity values during an activity.

see also lis:representedIn

scope note The 'is about' relation is intended as a superrelation for the broad range of relations between information objects and entities, including describing, naming, characterisation, and evaluation.

see also obo:IAO_0000136

comment IAO_0000136 is 'is about', in the Information Artifact Ontology.

related in ISO 15926 lis2:RepresentationOfThing

related in ISO 15926 lis12:representationByInformationObject

```
ObjectProperty: lis:isAbout
  Domain: lis:InformationObject
```

5.5.16.1 lis:represents

deprecated? true

comment Since we have the pair 'is about'/'represented in', it appears unnecessary to retain 'represents' and 'represented by' just for the link to ISO 15926-2.

see also lis:representedBy

usage note For annotations, see the inverse relation 'represented by'.

primitive? false

```
ObjectProperty: lis:represents
  InverseOf: lis:representedBy
```

5.5.16.2 lis:quantifiesQuality

see also lis:qualityQuantifiedAs

usage note For annotations, see the inverse relation 'quality quantified as'.

primitive? false

```
ObjectProperty: lis:quantifiesQuality
  InverseOf: lis:qualityQuantifiedAs
```

5.5.16.3 lis:quantityDatumOfProfile

see also lis:profileQuantityDatum

usage note For annotations, see the inverse relation 'profile quantity datum'.

primitive? false

```
ObjectProperty: lis:quantityDatumOfProfile
  InverseOf: lis:profileQuantityDatum
```

5.5.17 lis:representedIn

see also lis:isAbout

usage note For annotations, see the inverse relation 'is about'.

primitive? false

```
ObjectProperty: lis:representedIn
  Range: lis:InformationObject
  InverseOf: lis:isAbout
```

5.5.17.1 lis:representedBy

deprecated? true

comment Since we have the pair 'is about'/'represented in', it appears unnecessary to retain 'represents' and 'represented by' just for the link to ISO 15926-2.

comment In ISO 15926-2, this is the top-level relation from things to information objects. For IDO, we introduce representedIn, as the more generic inverse of isAbout.

equivalent in ISO 15926 lis2:RepresentationOfThing

equivalent in ISO 15926 lis12:representationByInformationObject

```
ObjectProperty: lis:representedBy
  Range: lis:InformationObject
```

5.5.17.2 lis:qualityQuantifiedAs

TODO. Clarify how to use category codes, such as colour names.

definition If x is a ‘quality quantified as’ y, then x is an ‘quality’ that is ‘represented in’ the ‘quality datum’ y.

primitive? false

example The nominal outer diameter of a pipe is 324 mm, the measured temperature of a boiler is 150 degrees Celsius, the colour of a surface is ‘red’.

see also lis:quantifiesQuality

usage note The range of ‘quality quantified as’ is ‘quality datum’, which allows for data that don’t follow the “unit of measure with value” pattern. Examples are colours and chemical composition categories. TODO To be clarified.

see also obo:IAO_0000417

comment IAO_0000417 is ‘is quality measured as’ of the Information Artefact Ontology. In IDO, the term “quantified” replaces “measured” to support cases where measurement is not involved, as in, e.g., estimates.

see also ssn:observedProperty

related in ISO 15926 lis2:PropertyQuantification

related in ISO 15926 lis12:PhysicalProperty

related in ISO 15926 lis12:QuantityMeasure

```
ObjectProperty: lis:qualityQuantifiedAs
  Domain: lis:Quality
  Range: lis:QualityDatum
```

EXAMPLE. A *Heat Exchanger* has a temperature that is represented by two timestamped *Datum* individuals.

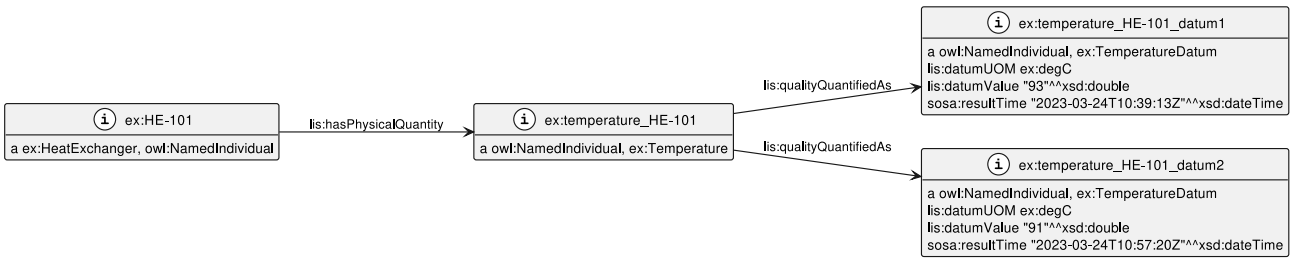


Figure 5.26 — ‘quality quantified as’ example

5.5.17.3 lis:profileQuantityDatum

definition If x has ‘profile quanity datum’ y, then x is an ‘activity profile’, y is a ‘quality datum’, and at a time within the duration of x, y quantifies [TODO. rewrite with a different term than “quantifies”] a quality of a participant in x.

primitive? true

why primitive TODO (revisit). The notion of “during” used here goes beyond OWL expressivity.

example The pressure of a liquid is measured at 10, 11, and 12 bar at different points of time during a transport activity.

see also lis:quantityDatumOfProfile

usage note Assume y is an ‘activity profile’, obtained from some containing activity z by selective abstraction on a quality q of a participant in z. Then, relate y by ‘profile quantity datum’ to magnitudes of q that are recorded during the execution of y.

comment Intuitively, ‘profile quantity datum’ should only point to quantity data recorded while the activity is taking place. This can not be enforced with OWL restrictions on the relation (it would

require a “diamond” pattern). However, with a suitable date or time data property value restriction, an individual activity can constrain related quantity data.

```
ObjectProperty: lis:profileQuantityDatum
  Domain: lis:ActivityProfile
  Range: lis:QualityDatum
```

5.5.18 lis:locatedRelativeTo

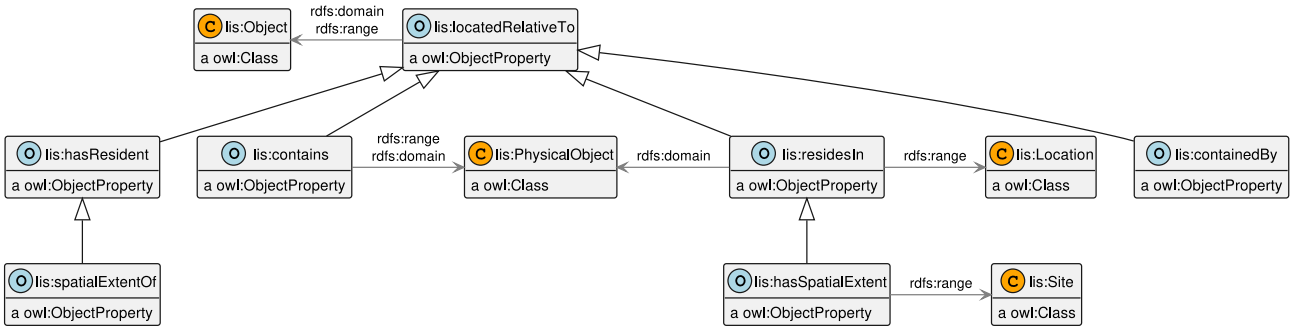


Figure 5.27 — IDO ‘located relative to’ relations hierarchy

definition If x is ‘located relative to’ y, then x and y are ‘object’s with location in space.

primitive? true

why primitive The notion of relative location in space is introduced with this relation.

example An outlet is below an inlet, a door is inside an opening, a motor is inside an enclosure.

usage note This relation is “abstract” (in the object-orientation sense of the word) in that it serves only as a parent relation to specialized relations of relative location. [TODO improve; any object has location in space, so the definition is trivial.]

equivalent in ISO 15926 lis2:RelativeLocation

equivalent in ISO 15926 lis12:locatedRelativeTo

see also obo:BFO_0000171

comment BFO_0000171 is ‘located in at some time’ in BFO-ISO. This relation has domain and range restricted to ‘independent continuant’ and not ‘spatial region’, meaning it’s a relation between entities that are material objects or determined by material objects, only.

see also obo:BFO_0000210

comment BFO_0000210 is ‘occupies spatial region at some time’ in BFO-ISO. This relation has domain restricted to ‘independent continuant’ and not ‘spatial region’, and range to ‘spatial region’, meaning it’s a relation from entities that are material objects or determined by material objects, to spatial regions determined independently of material objects.

see also dol:generic-location

```
ObjectProperty: lis:locatedRelativeTo
  Domain: lis:Object
  Range: lis:Object
  Characteristics: Symmetric
```

5.5.18.1 lis:contains

definition If x ‘contains’ y, then x and y are ‘physical object’s, and y is located entirely within x.

primitive? true

why primitive The notion of being “located within” is not fully defined here, leaving open whether this is understood in terms of boundaries of physical objects or a fixed frame of reference.

example An instrument is contained in an enclosure, an pump contains an impeller, an device is contained by a volume of inert gas, a pipeline inspection gauge (pig) is located inside a pipeline.

see also lis:containedBy

usage note This relation is intended to be used where an object x is located inside another object y, but x not is not a part of y (as would be the case, e.g., for an assembly).

see also obo:BFO_0000124

comment BFO_0000124 is 'location of at some time' in BFO-ISO.

see also obo:RO_0001015

comment RO_0001015 is 'location of' in the Relations Ontology.

equivalent in ISO 15926 lis2:ContainmentOfIndividual

equivalent in ISO 15926 lis12:contains

```
ObjectProperty: lis:contains
  Domain: lis:PhysicalObject
  Range: lis:PhysicalObject
```

5.5.18.2 lis:containedBy

see also lis:contains

usage note For annotations, see the inverse relation 'contains'.

primitive? false

```
ObjectProperty: lis:containedBy
  InverseOf: lis:contains
```

5.5.18.3 lis:residesIn

definition If x 'resides in' y, then x is a 'physical object', y is a 'location', and x is located entirely within y.

primitive? true

why primitive The notion of being "located within" is not fully defined here, leaving open whether this is understood in terms of boundaries of physical objects or a fixed frame of reference.

example An engine is located in a machine room, a pressure vessel is located in an outdoors plant area.

see also lis:hasResident

usage note This relation may be used where an object x is located in a site y, defined in terms of boundaries of a physical object, that may change its location over time.

related in ISO 15926 lis2:RelativeLocation

related in ISO 15926 lis12:locatedRelativeTo

```
ObjectProperty: lis:residesIn
  Domain: lis:PhysicalObject
  Range: lis:Location
```

5.5.18.4 lis:hasSpatialExtent

NOTE. This relation was added on 2023-02-01.

definition If x 'has spatial extent' y, then x is a 'physical object', and y is the 'site' determined as the extension in space of x.

primitive? true

why primitive The notion of "spatial extension" an object is not fully defined.

example The volume occupied by an process skid, the volume occupied by a volume of flammable gas.

see also lis:spatialExtentOf

usage note The ‘site’ of an object will occupy different ‘spatial location’s over time if the location and size of the object changes.

see also obo:BFO_0000210

comment BFO_0000210 is ‘occupies spatial region at some time’ (from BFO-ISO) (paired with BFO_0000211 ‘occupies spatial region at all times’), with ‘spatial region’ as range, which is incompatible with ‘has spatial extent’ in IDO, where the range is ‘site’.

see also dol:exact-location

```
ObjectProperty: lis:hasSpatialExtent
  Range: lis:Site
  Characteristics: Functional
```

5.5.18.5 lis:hasResident

see also lis:residesIn

usage note For annotations, see the inverse relation ‘resides in’.

primitive? false

related in ISO 15926 lis2:RelativeLocation

related in ISO 15926 lis12:locatedRelativeTo

```
ObjectProperty: lis:hasResident
  InverseOf: lis:residesIn
```

5.5.18.6 lis:spatialExtentOf

This relation was added on 2023-02-28.

see also lis:hasSpatialExtent

usage note For annotations, see the inverse relation ‘has spatial extent’.

primitive? false

```
ObjectProperty: lis:spatialExtentOf
  InverseOf: lis:hasSpatialExtent
```

5.5.19 lis:hasPart

definition If x ‘has part’ y, then y is a part of x.

primitive? true

why primitive Parthood is a main primitive for ontologies.

example A system has functional parts, a process has stages, a plant is partitioned into areas.

see also lis:partOf

see also obo:BFO_0000051

comment BFO_0000051 is ‘has part’, in the Information Artefact Ontology and in the Relations Ontology (in the latter, BFO_0000051 appears as a subrelation of ‘overlaps’).

see also obo:BFO_0000117

comment BFO_0000117 is ‘has occurrent part’ in BFO-ISO.

see also obo:BFO_0000178

comment BFO_0000178 is ‘has continuant part of at some time’ in BFO-ISO.

see also dol:part

see also ssn:hasSubSystem

equivalent in ISO 15926 CompositionOfIndividual

equivalent in ISO 15926 lis12:hasPart

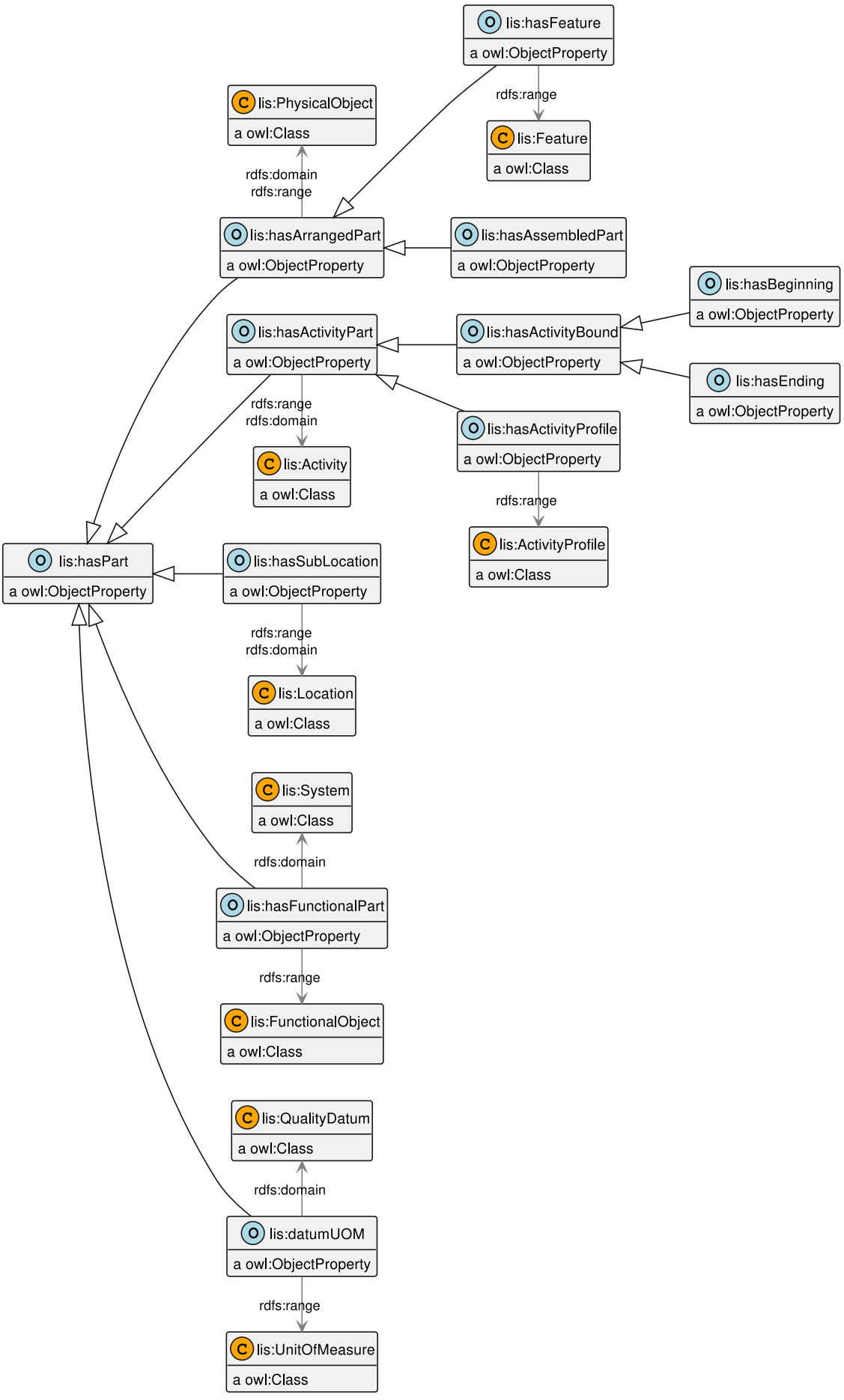


Figure 5.28 — IDO 'has part' relations hierarchy

EXAMPLE. A pressure gauge assembly is needed for installation.

A typical ‘pressure gauge assembly’ consists of a ‘pressure gauge’, a ‘piping tee’, two pieces of ‘threaded pipe’, and a ‘globe valve’. (The gauge is screwed into the branch end of the tee, the pipes are screwed into the headers of the tee, and the valve is screwed into one of the pipe ends.)

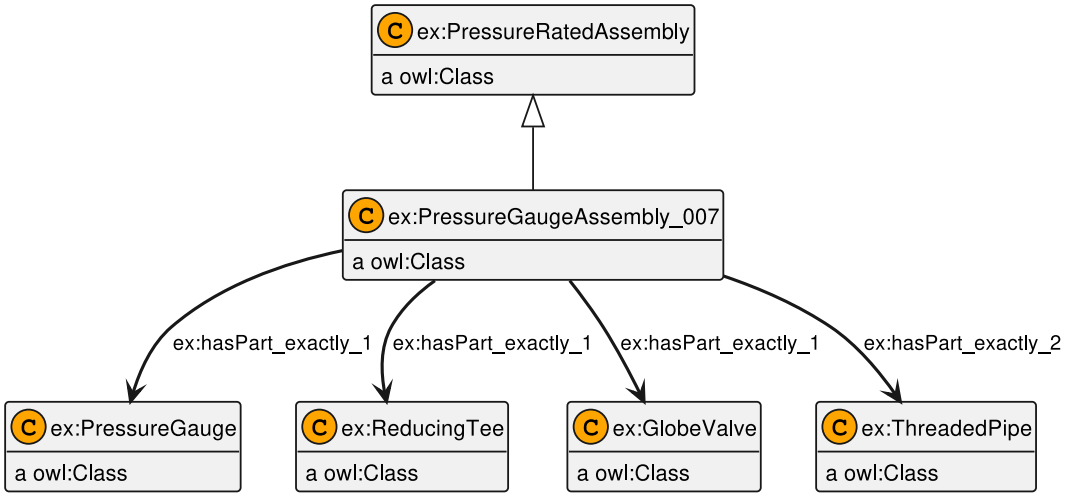


Figure 5.29 — ‘has part’ example, generic ‘pressure gauge assembly’ class

Plant conditions require compliance to the ASME B31.3 design code. We select ‘pressure gauge assembly 007 NPS 1’, a design which is represented in the ACME RDL as a subclass of ‘pressure rated assembly ASME B31.3’.

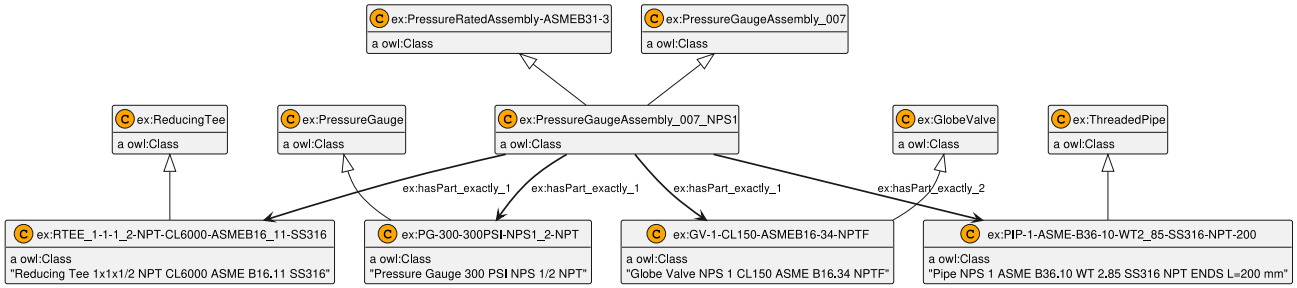


Figure 5.30 — ‘has part’ example, specialised ‘pressure gauge assembly’ class

An exemplar of the chosen design is assembled, represented as individual ‘P-12345-PA-01’. Note that if needed, each part of this individual could also be instantiated as an individual.

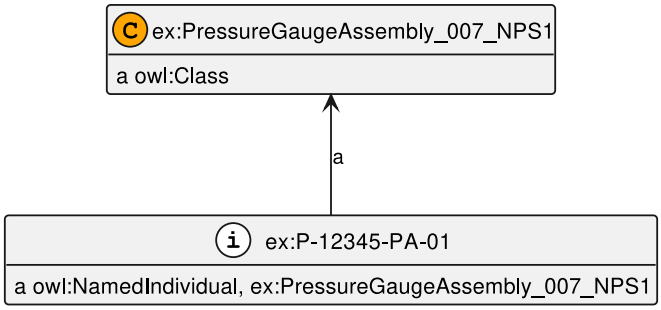


Figure 5.31 — ‘has part’ example, instance of specialised class

5.5.19.1 lis:hasFunctionalPart

definition If x ‘has functional part’ y, then x is a ‘system’ of which y is a part, and y has one or more functions that contribute to the function of x.

primitive? true

why primitive The notion of "contributing function" is not fully accounted for in IDO; a proper explanation might require an account of parts of functions themselves, i.e., of how simple functions can be composed into complex functions.

example A separation system has a cooling system as functional part.

see also lis:partOf

usage note 'has functional part' is used to represent the functional break-down of a system. Where x is a functional part of a system y, the realisation of one or more functions of x contributes to the performance of y.

```
ObjectProperty: lis:hasFunctionalPart
  Domain: lis:System
  Range: lis:FunctionalObject
```

5.5.19.2 lis:hasArrangedPart

definition If x 'has arranged part' y, then y is a part of x with a boundary determined in relation to other parts of x.

primitive? true

why primitive The notion of determinate boundary is taken as primitive. See the account of "fiat part" in BFO documentation.

example The upper half of a pressure vessel, the mid-section of a bridge.

see also lis:arrangedPartOf

see also obo:BFO_0000024

comment BFO_0000024 is 'fiat object part' in BFO-ISO.

equivalent in ISO 15926 lis2:ArrangementOfIndividual

equivalent in ISO 15926 lis12:hasArrangedPart

```
ObjectProperty: lis:hasArrangedPart
  Domain: lis:PhysicalObject
  Range: lis:PhysicalObject
```

5.5.19.3 lis:hasFeature

definition If x 'has feature' y, then y is a non-separable, contiguous 'arranged part' of x.

primitive? true

why primitive The notion of non-separable and contiguous part is primitive.

example A flange face, a corrosion protected surface, the contact surface of an electrical plug.

see also lis:featureOf

comment The definition is lifted directly from 'feature whole part ' of ISO 15926-2.

equivalent in ISO 15926 lis2:FeatureWholePart

equivalent in ISO 15926 lis12:hasFeature

```
ObjectProperty: lis:hasFeature
  Range: lis:Feature
```

5.5.19.4 lis:hasAssembledPart

definition If x 'has assembled part' y, then y is a connected 'arranged part' of x.

primitive? true

why primitive The notion of connection is primitive.

example A pipe bend welded into a piping spool, the protective sheath on an electrical cable, a door-handle in a door.

see also lis:assembledPartOf

equivalent in ISO 15926 lis2:AssemblyOfIndividual

equivalent in ISO 15926 lis12:hasAssembledPart

5.5.19.5 lis:hasActivityPart

definition If x 'has activity part' y, then x and y are activities, and x 'has arranged part' y.

primitive? false

example The startup period of a chemical process, the survey part of a maintenance activity, the moment a software program terminates execution.

see also lis:activityPartOf

scope note Parts of activities may be determined by a range of criteria, including temporal restrictions (to sub-periods), spatial restrictions (parts that occupy sub-volumes of space), by the engagement of participants (temporal or spatial parts where one or more participants are active), and "selective abstraction" to profiles (parts determined by that which affects variation in selected qualities).

see also obo:BFO_0000117

comment BFO_0000117 is 'has occurrent part' in BFO-ISO. This relation has 'occurrent' as domain and range, where IDO 'has activity part' is restricted to just 'activity'.

related in ISO 15926 lis2:TemporalWholePart

related in ISO 15926 lis12:hasTemporalPart

```
ObjectProperty: lis:hasActivityPart
  Domain: lis:Activity
  Range: lis:Activity
```

5.5.19.6 lis:hasActivityBound

definition If x 'has activity bound' y, then x 'has activity part' y, and y is an 'event'.

primitive? false

example The moment an electrical connection is broken, the time that a fluid reaches a set temperature.

see also lis:activityBoundOf

equivalent in ISO 15926 lis2:TemporalBounding

equivalent in ISO 15926 lis12:hasTemporalBound

5.5.19.7 lis:hasBeginning

definition If x 'has beginning' y, then y is an 'event' which 'has temporal extent' some 'instant' which is the first instant of the temporal extent of x.

primitive? true

why primitive The notion of first instant is not defined.

example The event that light is switched on, the start of filling a tank.

see also lis:begins

see also obo:BFO_0000222

comment BFO_0000222 is 'has first instant' in BFO-ISO, defined for 'temporal region' (differently from 'has beginning' in IDO, defined for 'activity').

equivalent in ISO 15926 lis2:Beginning

equivalent in ISO 15926 lis12:hasBeginning

5.5.19.8 lis:hasEnding

definition If x 'has ending' y, then y is an 'event' which 'has temporal extent' some 'instant' which is the last instant of the temporal extent of x.

primitive? true

why primitive The notion of last instant is not defined.

example The event that light is switched off, the end of filling a tank.

see also lis:ends

see also obo:BFO_0000224

comment BFO_0000224 is 'has last instant' in BFO-ISO, defined for 'temporal region' (differently from 'has ending' in IDO, defined for 'activity').

equivalent in ISO 15926 lis2:Ending

equivalent in ISO 15926 lis12:hasEnd

5.5.19.9 lis:hasActivityProfile

definition If x 'has activity profile' y, then x 'has activity part' y, and y is determined by selective abstraction to that part of x that effects variation in some quality of a participant in x.

primitive? true

why primitive The notion of selective abstraction is not defined.

example A pipeline transport activity has a "profile" part which is the variation in fluid pressure.

see also lis:activityProfileOf

see also obo:BFO_0000133

comment BFO_0000133 is 'process profile of', in the Molecular Process Ontology and other BFO-based ontologies.

```
ObjectProperty: lis:hasActivityProfile
#  Domain: lis:Activity
  Range: lis:ActivityProfile
```

5.5.19.10 lis:hasSubLocation

definition If x 'has sublocation' y, then x and y are locations, and x 'has part' y.

primitive? false

example The landing area is a sublocation of the helicopter deck.

see also lis:subLocationOf

```
ObjectProperty: lis:hasSubLocation
  Domain: lis:Location
  Range: lis:Location
```

5.5.19.11 lis:datumUOM

definition If x 'has datum uom' y, then x is a 'quality datum' for which any 'datum value' is interpreted by the 'unit of measure' y.

primitive? true

why primitive Interpretation in terms of a unit is not defined.

example The measured datum "55 degrees Celsius" has Celsius as unit of measure.

see also lis:uomOfDatum

see also om:hasUnit

see also http://qudt.org/schema/qudt/unit

see also obo:IAO_0000039

comment IAO_0000039 is 'has measurement unit label' in the Information Artifact Ontology.

see also https://schema.org/unitCode

see also obo:RO_0002536

comment RO_0002536 is 'measurement property has unit', an annotation property in the Relations Ontology.

```
ObjectProperty: lis:datumUOM
  Domain: lis:QualityDatum
  Range: lis:UnitOfMeasure
  Characteristics: Functional
```

5.5.20 lis:partOf

see also lis:hasPart

usage note For annotations, see the inverse relation 'has part'.

primitive? false

```
ObjectProperty: lis:partOf
  InverseOf: lis:hasPart
```

5.5.20.1 lis:functionalPartOf

see also lis:hasFunctionalPart

usage note For annotations, see the inverse relation 'has functional part'.

primitive? false

```
ObjectProperty: lis:functionalPartOf
  InverseOf: lis:hasFunctionalPart
```

5.5.20.2 lis:arrangedPartOf

see also lis:hasArrangedPart

usage note For annotations, see the inverse relation 'has arranged part'.

primitive? false

```
ObjectProperty: lis:arrangedPartOf
  InverseOf: lis:hasArrangedPart
```

5.5.20.3 lis:featureOf

see also lis:hasFeature

usage note For annotations, see the inverse relation 'has feature'.

primitive? false

```
ObjectProperty: lis:featureOf
  InverseOf: lis:hasFeature
```

5.5.20.4 lis:assembledPartOf

see also lis:hasAssembledPart

usage note For annotations, see the inverse relation 'has assembled part'.

primitive? false

```
ObjectProperty: lis:assembledPartOf
  InverseOf: lis:hasAssembledPart
```

5.5.20.5 lis:activityPartOf

see also lis:hasActivityPart

usage note For annotations, see the inverse relation ‘has activity part’.

primitive? false

```
ObjectProperty: lis:activityPartOf
  InverseOf: lis:hasActivityPart
```

5.5.20.6 lis:activityBoundOf

see also lis:hasActivityBound

usage note For annotations, see the inverse relation ‘has activity bound’.

primitive? false

```
ObjectProperty: lis:activityBoundOf
  InverseOf: lis:hasActivityBound
```

5.5.20.7 lis:begins

see also lis:hasBeginning

usage note For annotations, see the inverse relation ‘has beginning’.

primitive? false

```
ObjectProperty: lis:begins
  InverseOf: lis:hasBeginning
```

5.5.20.8 lis:ends

see also lis:hasEnding

usage note For annotations, see the inverse relation ‘has ending’.

primitive? false

```
ObjectProperty: lis:ends
  InverseOf: lis:hasEnding
```

5.5.20.9 lis:activityProfileOf

see also lis:hasActivityProfile

usage note For annotations, see the inverse relation ‘has activity profile’.

primitive? false

```
ObjectProperty: lis:activityProfileOf
  InverseOf: lis:hasActivityProfile
```

5.5.20.10 lis:subLocationOf

see also lis:hasSubLocation

usage note For annotations, see the inverse relation ‘has sub location’.

primitive? false

```
ObjectProperty: lis:subLocationOf
  InverseOf: lis:hasSubLocation
```

5.5.20.11 lis:uomOfDatum

see also lis:datumUOM

usage note For annotations, see the inverse relation ‘datumUOM’.

primitive? false

comment Relation from ‘unit of measure’ to ‘quantity datum’. The inverse relation of datumUOM.

```
ObjectProperty: lis:uomOfDatum
  InverseOf: lis:datumUOM
```

5.5.21 lis:hasParticipant

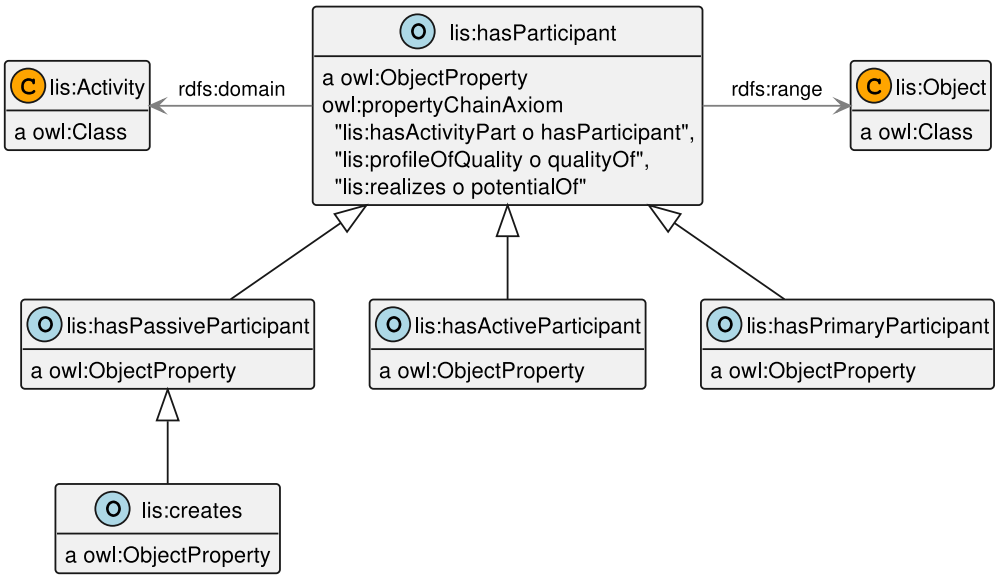


Figure 5.32 — IDO 'has participant' relations hierarchy

definition If x 'has participant' y, then x is an 'activity', and y is an 'object' that participates in x.

primitive? true

why primitive The notion of participation is primitive.

example A pump participates in (a) pumping (activity), an electric transformer participates in voltage conversion, a stream of water participates in dilution, a bolt participates in a connection.

see also lis:participatesIn

see also obo:BFO_0000057

comment BFO_0000057 is 'has participant at some time'.

see also obo:RO_0000057

comment RO_0000057 is 'has participant'.

comment A property chain axiom secures that a participant in part of an activity participates in the activity as a whole.

equivalent in ISO 15926 lis2:Participation

equivalent in ISO 15926 lis12:hasParticipant

```
ObjectProperty: lis:hasParticipant
  Domain: lis:Activity
  Range: lis:Object
  SubPropertyChain:
    lis:hasActivityPart o lis:hasParticipant
```

usage note A property chain axiom secures that when a realizable (a 'disposition' or 'role') is 'realized in' an activity, the activity has the bearer as a participant. [TODO required participation may not be appropriate for 'interest's]

```
ObjectProperty: lis:hasParticipant
  SubPropertyChain:
    lis:realizes o lis:potentialOf
```

usage note A property chain axiom secures that when an activity has a 'profile quantity datum', the bearer of the quantity is a participant in the activity.

```
ObjectProperty: lis:hasParticipant
  SubPropertyChain:
    lis:profileOfQuality o lis:qualityOf
```

EXAMPLE. [TODO Provide an example that uses IDO to represent an IDEF0 function model.]

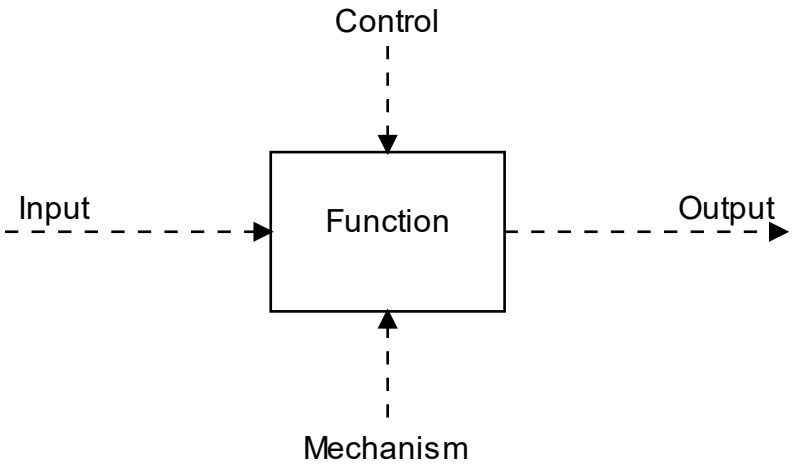


Figure 5.33 — 'has participant' example, IDEF0 Box Format

Source for IDEF0: Defense Acquisition University Press. "System Engineering Fundamentals." January 2001. Available from https://ocw.mit.edu/courses/16-885j-aircraft-systems-engineering-fall-2005/pages/readings/.

5.5.21.1 lis:hasActiveParticipant

NOTE. Tentative addition [2023-02-01].

definition If x 'has active participant' y, then x is an active participant in y.

primitive? true

why primitive The distinction between active and participation participation is primitive.

example A hammer is the tool in a nail-driving activity, a pressure transmitter outputs data in a monitoring activity, a maintenance engineer replaces an instrument in a repair activity, a light sensor controls the level of illumination.

see also lis:activeParticipantIn

usage note This is a covering relation for participation that involves acting on other participants.

usage note With OWL property chains, the right kind of participation can be secured for activities that count as realizations of 'potential' individuals. As in, a pump's primary function is realized in pumping activities, but only provided the pump is actually doing the pumping, acting as an "agent" or "subject".

5.5.21.2 lis:hasPassiveParticipant

NOTE. Tentative addition [2023-02-01].

definition if x 'has passive participant' y, then x is a passive participant in y.

primitive? true

why primitive The distinction between active and participation participation is primitive.

example A metal plate is perforated in a drilling activity, a volume of liquid is heated in a heat transmission activity, a loading ramp is the location of an offloading activity, a fluid medium is pumped, a piece of metal is shaped, a stream of data is filtered.

see also lis:passiveParticipantIn

usage note This is a covering relation for passive participation, i.e., being acted on by other participants in an activity, or playing a supporting role.

usage note With OWL property chains, the right kind of participation can be secured for activities that count as realizations of 'potential' individuals. As in, a bolt is a passive participant in a bolt-driving activity, where its disposition to enter a threaded hole is realized (distinct from the bolt's function, which is realized in a subsequent activity of holding pieces together).

5.5.21.3 lis:creates

Changes from Part 14:

This relation has been relocated from a top-level relation to a subrelation of 'has passive participant'.

Tentatively, the range of this relation has been made more general, replacing 'physical object' with 'object'; this allows the relation to be applied to creation of not just artefacts, but also information objects, organizations, and locations (in the latter case, 'site's, as would be created by the creation of physical boundaries).

Replace the range of 'physical object' with 'object' to allow for use with creation of non-physical objects.

definition if x 'creates' y, then x is an 'activity', y is an 'object', and x brings y into being.

primitive? true

why primitive A comprehensive definition of the notion of creation is beyond the scope of IDO.

example A manufacturing activity produces a car, a welding activity produces a piping spool, a measurement activity produces a performance log.

see also lis:createdBy

comment Use this relation to express that an activity brings an object into being.

related in ISO 15926 lis2:ClassOfCauseOfBeginningOfClassOfIndividual

comment ISO 15926-2, p. 132 provides an example as, "A car manufacturing activity causes the beginning of a car").

equivalent in ISO 15926 lis12:causesBeginningOf

5.5.21.4 lis:hasPrimaryParticipant

NOTE. Tentative addition *[2023-02-06 Mon]*. This relation is intended to capture participation of that object which "defines" the activity, understood as: The activity is a realization of a function of the primary participant, and no (non-temporal) part qualifies as a realization. (Alternatively, replace "function" with "role", "disposition", or "capability".)

definition If x 'has primary participant' y, then x is a realization of a disposition or role of y, and this realization serves to identify y [TODO improve].

primitive? true

why primitive The notion of primary participant is not defined.

example A pump is the performer in a pumping activity, a thermometer is the performer in a temperature measurement.

see also lis:primaryParticipantIn

usage note Assume p is a pump, a an activity that realizes the function fp of p, and no subactivity x of a realizes fp (except for subactivities determined solely by restriction to a temporal subinterval). For instance, the subactivity of the motor that drives the impeller realizes a function of the motor, but this is not by itself a realization of the function of the pump as a whole, hence the motor is not a 'primary participant'.

5.5.22 lis:participantIn

see also lis:hasParticipant

usage note For annotations, see the inverse relation ‘has participant’.

primitive? false

```
ObjectProperty: lis:participantIn
  InverseOf: lis:hasParticipant
```

5.5.22.1 lis:activeParticipantIn

see also lis:hasActiveParticipant

usage note For annotations, see the inverse relation ‘has active participant’.

primitive? false

```
ObjectProperty: lis:activeParticipantIn
  InverseOf: lis:hasActiveParticipant
```

5.5.22.2 lis:passiveParticipantIn

see also lis:hasPassiveParticipant

usage note For annotations, see the inverse relation ‘has passive participant’.

primitive? false

```
ObjectProperty: lis:passiveParticipantIn
  InverseOf: lis:hasPassiveParticipant
```

5.5.22.3 lis:createdBy

see also lis:creates

usage note For annotations, see the inverse relation ‘creates’.

primitive? false

```
ObjectProperty: lis:createdBy
  InverseOf: lis:creates
```

5.5.22.4 lis:primaryParticipantIn

see also lis:hasPrimaryParticipant

usage note For annotations, see the inverse relation ‘has primary participant’.

primitive? false

```
ObjectProperty: lis:primaryParticipantIn
  InverseOf: lis:hasPrimaryParticipant
```

5.5.23 lis:occursRelativeTo

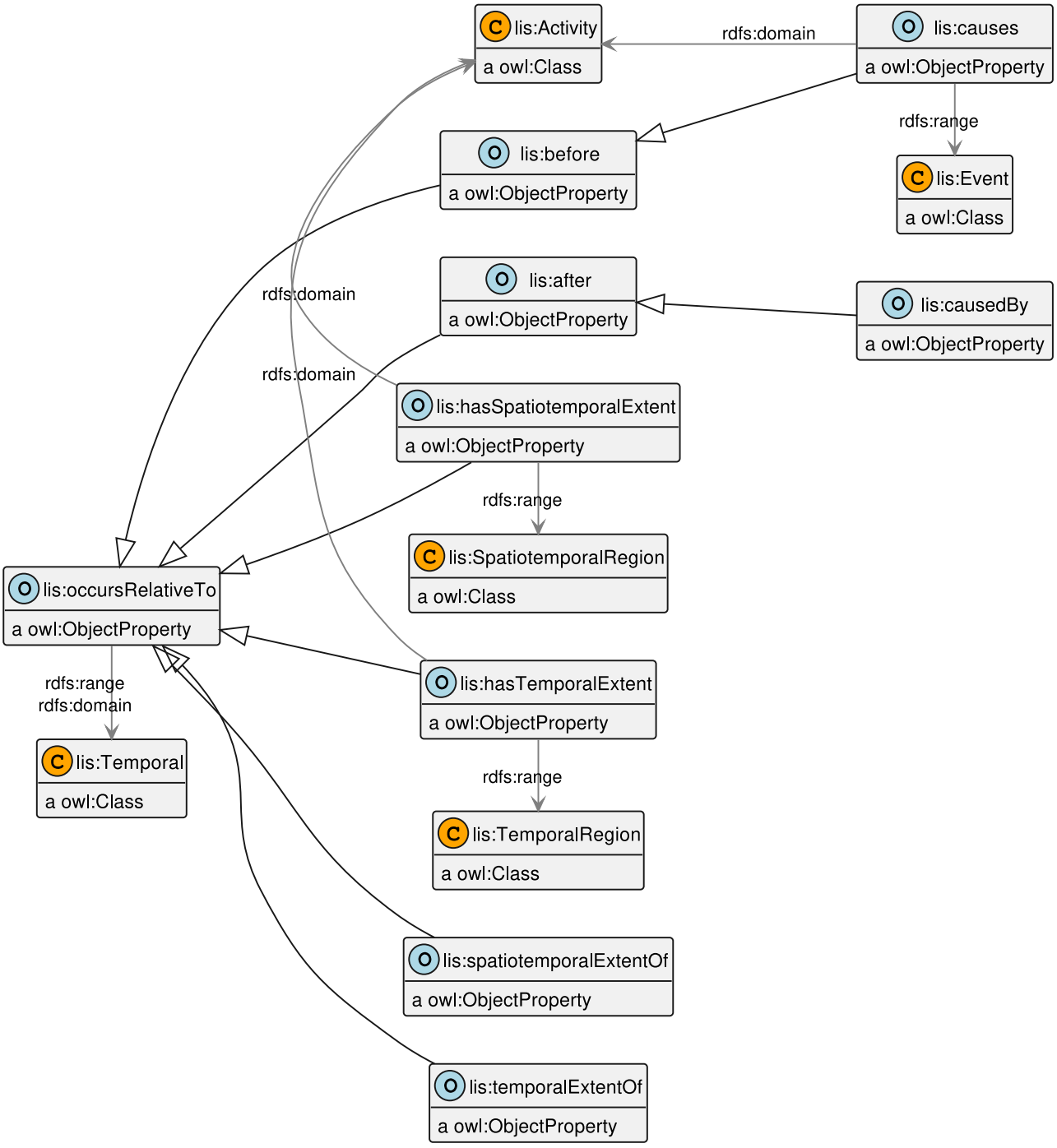


Figure 5.34 — IDO 'occurs relative to' relations hierarchy

definition If x 'occurs relative to' y, then x and y are 'temporal' entities, with temporal location.

primitive? true

why primitive The notion of relative location in time is introduced with this relation.

example A vessel is emptied before inspection, a valve is opened before draining of a pipeline, firmware is installed before the equipment begins to operate.

usage note This relation is "abstract" (in the object-orientation sense of the word) in that it serves only as a parent relation to specialised relations of relative temporal location. [TODO improve; any 'temporal' has temporal location, so the definition is trivial.]

comment This relation is introduced for the DL profile as a top relation for various temporal relations between activities (only).

remodels ISO 15926 lis2:TemporalSequence

```
ObjectProperty: lis:occursRelativeTo
  Domain: lis:Temporal
  Range: lis:Temporal
  Characteristics: Symmetric
```

5.5.23.1 lis:hasTemporalExtent

NOTE. Tentative addition *[2023-02-14 Tue]*.

definition If x 'has temporal extent' y, then x is an activity, and y is a temporal region which is the temporal extent of x.

primitive? true

why primitive The notion of temporal extension is primitive.

example A well drilling activity takes place in a 20-hour interval, a control system is switched on at 10:00.

see also lis:temporalExtentOf

see also BFO_0000199

comment BFO_0000199 is 'occupies temporal region'.

```
ObjectProperty: lis:hasTemporalExtent
  Domain: lis:Activity
  Range: lis:TemporalRegion
```

5.5.23.2 lis:temporalExtentOf

see also lis:hasTemporalExtent

usage note For annotations, see the inverse relation 'has temporal extent'.

primitive? false

```
ObjectProperty: lis:temporalExtentOf
  InverseOf: lis:hasTemporalExtent
```

5.5.23.3 lis:hasSpatiotemporalExtent

NOTE. Tentative addition *[2023-02-14 Tue]*.

definition If x 'has spatiotemporal extent' y, then x is an activity, and y is a spatiotemporal region which is the spatiotemporal extent of x.

primitive? true

why primitive The notion of spatiotemporal extension is primitive.

example A pipeline pig occupies the spatiotemporal region which is its trajectory (adopted from BFO-ISO).

see also lis:

see also obo:BFO_0000200

comment BFO_0000200 is 'occupies spatiotemporal region'.

```
ObjectProperty: lis:hasSpatiotemporalExtent
  Domain: lis:Activity
  Range: lis:SpatiotemporalRegion
```

5.5.23.4 lis:spatiotemporalExtentOf

NOTE. Tentative addition *[2023-03-02 Thu]*.

see also lis:hasSpatiotemporalExtent

usage note For annotations, see the inverse relation 'has spatiotemporal extent'.

primitive? false

```
ObjectProperty: lis:spatiotemporalExtentOf
  InverseOf: lis:hasSpatiotemporalExtent
```

5.5.23.5 lis:before

See Alignment with the Time Ontology in OWLfor a way to include a full set of temporal relations in IDO.

TODO. Clarify whether 'before' means not just starting before, but also ending before. This is left open in the text of ISO 15926-2.

definition If x is 'before' y, then x ends no later than the start of y.

primitive? true

why primitive The temporal vocabulary is primitive.

example The cooling stage takes place before separation.

see also lis:after

see also obo:BFO_0000063

comment BFO_0000063 is 'precedes' in BFO-ISO.

related in ISO 15926 lis2:ClassOfTemporalSequence

5.5.23.6 lis:causes

deprecated? true

comment This relation is too specialised to warrant inclusion in IDO.

definition If x is 'causes' y, then x is an 'activity' that is 'before' the 'event' y, and x causes y to happen.

primitive? true

why primitive The temporal vocabulary is primitive.

example The tanker loading activity caused the event described as 'tank liquid level full' (example from ISO 15926-2, p. 127)

see also lis:causedBy

equivalent in ISO 15926 lis2:CauseOfEvent

equivalent in ISO 15926 lis12:causes

```
ObjectProperty: lis:causes
  Domain: lis:Activity
  Range: lis:Event
```

5.5.23.7 lis:after

see also lis:before

usage note For annotations, see the inverse relation 'before'.

primitive? false

see also obo:BFO_0000062

comment BFO_0000062 is 'preceded by' in BFO-ISO.

related in ISO 15926 lis2:ClassOfTemporalSequence

```
ObjectProperty: lis:after
  InverseOf: lis:before
```

comment Use this relation to state that one activity occurs after another.

5.5.23.8 lis:causedBy

deprecated? true

comment This relation is too specialised to warrant inclusion in IDO.

see also lis:causes

usage note For annotations, see the inverse relation 'causes'.

primitive? false

```
ObjectProperty: lis:causedBy
  InverseOf: lis:causes
```

5.6 Data relations (data properties)

```
## Data property details
```

5.6.1 lis:datumValue

definition If x 'datum value' y, then x is a 'quality datum', and y is the literal value of x.

primitive? true

why primitive The notion of literal value is primitive.

example The measured datum "55 degrees Celsius" has "55" as literal value.

usage note A literal value is uninterpreted. Provide a unit of measure, or other category, for a 'quality datum' to be interpretable.

usage note For numeric non-integer values, the recommended data type is xsd:float.

see also http://qudt.org/schema/qudt/numericValue

comment numericValue is a data property in QUDT.

see also om:hasNumericalValue

comment hasNumericalValue is a data property in the Ontology of Units of Measure.

see also ssn:hasSimpleResult

comment hasSimpleResult is a data property in the Semantic Sensor Networks ontology.

comment sosa:Observation is akin to lis:Activity, not to lis:QuantityDatum (see https://www.w3.org/TR/vocab-ssn/#SOSAresultTime)/ sosa:Result is a subclass of lis:QuantityDatum [TODO also see lis:timestamp]

see also obo:IAO_0000004

comment IAO_0000004 is the data property 'has measurement value', in the Information Artefact Ontology.

comment BFO-ISO declares no data properties.

```
DataProperty: lis:datumValue
 Domain: lis:QualityDatum
 Characteristics: Functional
```

5.6.2 lis:qualityQuantityValue

definition If x has 'quality quantity value' y, then x is an 'object', and y is the 'datum value' of a 'quality datum' that 'quantifies quality' a 'quality' of x.

primitive? false

example Mass in kilograms, pressure in barg.

usage note This relation is "abstract" (in the object-orientation sense of the word) in that it serves only as a parent relation to specialised relations. It is intended to serve as a super-property for "shortcut" relations that combine a quality, lis:qualityQuantifiedAs, and a unit of measure into a simple data property.

```
DataProperty: lis:qualityQuantityValue
 Domain: lis:Object
```

5.6.3 lis:timestamp

NOTE. Tentative addition [MMB 2023-03-13].

[TODO Comments to be worked into the text: This is needed in use cases that represent measurements related to events, e.g. a CT scan event will be described by a timestamp (more precisely start and end timestamp) and the measurements such as voltage, amperage, etc. Without a data property for timestamps, we would have to generate instant objects for each event. Moreover, automatically generated machine events (such as temperature exceeding a certain range) have timestamps too, and a data property would save us a generation of many Instants.]

[TODO. Versus xsd:dateTime, xsd:dateTimeStamp makes the time zone expression obligatory (according to http://www.datypic.com/sc/xsd11/t-xsd_dateTimeStamp.html). Can we make do with the more common xsd:dateTime?]

[TODO. There's a need for timestamping at least (1) event start and end times, and (2) quantity datum individuals. This must be reflected in appropriate rdfs:domain restrictions. We likely need two, or more, data properties; reuse from OWL Time may be recommended.]

comment The Time Ontology https://www.w3.org/TR/owl-time/ introduces several data properties to annotate Instants with their time positions (including time:inXSDDateTimeStamp etc). The lis:timestamp property allows to directly add timestamps to events, without resorting to Instant objects that are temporal extents of an event. We may also add two subproperties lis:startTimestamp and lis:endTimestamp.

```
DataProperty: lis:timestamp
 Domain: lis:Activity
 Range: xsd:dateTimeStamp
```

5.6.4 lis:approvedOn

deprecated? true

comment This is a super-property for stating the time that an entity was approved, derived from Part 2 "approval". Introduce sub-properties to match different contexts and types of approval. The range of sub-properties should be xsd:date or xsd:dateTime.

related in ISO 15926 lis2:Approval

5.7 Annotation relations (annotation properties)

```
## Annotation property details
```

5.7.1 rdfs:comment

defined by http://www.w3.org/2000/01/rdf-schema#

deprecated? true

explanatory note The IOF Annotation Property Guide V2.2 states, "comment MUST NOT be used".

5.7.2 rdfs:label

defined by http://www.w3.org/2000/01/rdf-schema#

5.7.2.1 skos:prefLabel

comment Preferred label for the ontology resource.

defined by http://www.w3.org/2004/02/skos/core

5.7.2.2 skos:altLabel

comment Alternative label for the ontology resource.

defined by http://www.w3.org/2004/02/skos/core

5.7.2.3 iof-av:synonym

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.3 see also (rdfs:seeAlso)

defined by http://www.w3.org/2000/01/rdf-schema#

5.7.3.1 rdfs:isDefinedBy

defined by http://www.w3.org/2000/01/rdf-schema#

5.7.4 lis:originatesFrom

comment This is a super-property for associating a resource with a resource it originated from.

5.7.4.1 lis:transformedFrom

comment This annotation property is used for stating that the current resource originates from another resource by transformation.

5.7.4.2 lis:mergedFrom

comment This annotation property is used for stating that the current resource originates from one or several other resources by merging.

5.7.4.3 lis:splitFrom

comment This annotation property is used for stating that the current resource originates from another resource by splitting.

5.7.5 lis:relatedEntity

comment Annotation property for referring from OWL ontology resources to entities in external vocabularies (typically, but not necessarily, ontologies), where definitions, examples, or explications of the external entity contribute to understanding and defining the local resource.

5.7.5.1 lis:relatedEntityISO15926

comment Annotation property for referring from OWL ontology resources in ISO 15926-14 to entities in other parts of ISO 15926, where definitions, examples, or explications of the Part 2 entity contribute to understanding and defining the Part 14 resource.

example The Part 14 class Activity is clearly related to, but not equivalent to, the Part 2 entity ClassOfActivity, and this annotation property should be used to record that fact.

5.7.5.2 lis:remodelsEntity

comment Annotation property for referring from OWL ontology resources to entities in external vocabularies, where the local resource, together with OWL constructs, can express intuitively the same facts as the referenced entity.

5.7.5.3 lis:remodelsEntityISO15926

comment Annotation property for referring from OWL ontology resources in ISO 15926-14 to entities in other parts of ISO 15926 (primarily Part 2), where the Part 14 entity type, together with OWL constructs, can express intuitively the same facts as the referenced entity.

example Range restrictions on physical quantities can be captured with OWL and Part 14 using "facet" datatype restrictions. These will correspond to Part 2 Property Range expressions.

```
AnnotationProperty: lis:remodelsEntityISO15926
  SubPropertyOf: lis:relatedEntityISO15926
```

5.7.5.4 lis:equivalentEntity

comment Annotation property for referring from OWL ontology resources to entities in external vocabularies, where the local resource is equivalent in meaning to that of the referenced entity.

5.7.5.5 lis:equivalentEntityISO15926

comment Annotation property for referring from ontology resources in ISO 15926-14 to entities in other parts of ISO 15926, where the Part 14 resource is equivalent in meaning to that of the referenced entity.

example The Part 14 class Activity has, intuitively, the same meaning as the Part 2 entity Activity, and this annotation property should be used to record that fact.

```
AnnotationProperty: lis:equivalentEntityISO15926
  SubPropertyOf: lis:relatedEntityISO15926
```

5.7.5.6 lis:deprecatedEntity

comment Annotation property for referring from OWL ontology resources to semantically similar entities in external vocabularies, where the limitations of OWL, as as first-order language, imply that representing the intended meaning of the referenced entity is not feasible.

5.7.5.7 lis:deprecatedEntityISO15926

comment Annotation property for referring from ontology resources in ISO 15926-14 to semantically similar entities in other parts of ISO 15926 (primarily Part 2), where the limitations of OWL, as as first-order language, imply that Part 14 is not capable of expressing the intended meaning of the referenced entity.

example Modal notions can not be captured in OWL. Therefore, Part 2 entities that are essentially modal, such as those referring to “actuality” or “materialisation”, need to be left out of Part 14.

```
AnnotationProperty: lis:deprecatedEntityISO15926
  SubPropertyOf: lis:relatedEntityISO15926
```

5.7.6 iof-av:isPrimitive

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.7 skos:note

defined by http://www.w3.org/2004/02/skos/core

5.7.7.1 skos:definition

defined by http://www.w3.org/2004/02/skos/core

5.7.7.2 iof-av:naturalLanguageDefinition

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.7.3 iof-av:firstOrderLogicDefinition

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.7.4 iof-av:semiFormalNaturalLanguageDefinition

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.7.5 skos:example

defined by http://www.w3.org/2004/02/skos/core

5.7.7.6 iof-av:explanatoryNote

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.7.7 skos:scopeNote

defined by http://www.w3.org/2004/02/skos/core

5.7.7.8 iof-av:usageNote

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.7.9 iof-av:primitiveRationale

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.7.10 iof-av:semiFormalNaturalLanguageAxiom

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.8 pav:previousVersion

defined by http://purl.org/pav/

5.7.9 pav:derivedFrom

defined by http://purl.org/pav/

5.7.10 pav:lastUpdateOn

defined by http://purl.org/pav/

5.7.11 dcterms:source

defined by http://purl.org/dc/terms/

5.7.11.1 iof-av:adaptedFrom

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.11.2 iof-av:directSource

defined by https://spec.industrialontologies.org/ontology/core/meta/AnnotationVocabulary/

5.7.12 dcterms:title

defined by http://purl.org/dc/terms/

5.7.13 dcterms:license

defined by http://purl.org/dc/terms/

5.7.14 dcterms:description

defined by http://purl.org/dc/terms/

5.7.15 dcterms:issued

defined by http://purl.org/dc/terms/

5.7.16 dcterms:contributor

defined by http://purl.org/dc/terms/

5.7.16.1 dcterms:creator

defined by http://purl.org/dc/terms/

5.7.16.2 pav:createdBy

defined by http://purl.org/pav/

5.7.16.3 pav:contributedBy

defined by http://purl.org/pav/

5.7.17 dcterms:modified

defined by http://purl.org/dc/terms/

5.7.18 dcterms:publisher

defined by http://purl.org/dc/terms/

5.7.19 dc:rights

defined by http://purl.org/dc/elements/1.1/

5.7.20 foaf:isPrimaryTopicOf

defined by http://xmlns.com/foaf/spec/

6 Intended use of IDO

IDO is intended to be suitable for industrial use cases, to create vocabularies and asset models that employ the ontological vocabulary and exploit OWL DL reasoning for quality assurance and inference of implicit knowledge.

In typical applications, this ontology will be the ultimate node in an owl:imports hierarchy of ontologies.

[TODO Check for overlap between the following proposed sections, informative annexes, and the "modelling facts" chapter]

6.1 Asset model ontologies

This section covers best practices for asset model ontology modules and the use of OWL imports in asset models, facilitating the exchange of meaningful data between organisations, with a focus on class hierarchy depth.

6.2 Reference data libraries

This section discusses the recommended use of reference data in libraries based on IDO and the required content to enable semantic reasoning. It also provides recommendations on ontology modules and the use of OWL imports.

6.3 Locations

This section addresses area identification and classification for plant areas, management of overlapping plant areas, and the recommended use of residesIn and hasResident relationships (often many-to-many). It also covers the suggested use of hasSubLocation, primarily for hierarchical area breakdowns.

6.4 Barriers

This section explores the handling of barriers in an IDO-based asset model, applicable to both process design and barriers between plant areas.

6.5 Streams

his section provides best practices for managing streams, with components and parts.

6.6 Design conditions

This section discusses the recommended approach to handling design condition data related to areas, streams, specified equipment (TAG), product types, and product individuals.

6.7 Asset model change management

This section offers best practices for determining when a new specified equipment (TAG) individual is needed, based on changes in form, fit, or function.

6.8 Mapping existing data to IDO

[TODO Give an outline of best practice for mapping existing data to IDO.]

7 Modelling facts with IDO

[TODO This section has not been updated to reflect changes from the previous version of the ontology, so some terms will be misleading. For example, PeriodInTime is now called Interval. Etc.]

The following sections are for ontology practitioners.

What is this (for the editors)

It makes sense to provide expository sections describing bundles of related ontology resources. While we need the ontology to have definitions, examples, etc., for each individual resource, we also need to explain how classes and relations are intended to work together.

Such explanations will in many cases start from discussing selected relations.

It will be useful to provide pointers to relevant literature (papers, ontology development textbooks).

The following sections provide a start for some possible topics.

7.1 Independent and Dependent objects

The stated purpose of IDO is to be a reliable, basic vocabulary for facts about industrial plants. In practice, this starts with descriptions of the industrial artefacts – from components and equipment assemblies to whole factories. There are two other main types of entity that also need to be managed, namely, information objects and processes, to be discussed in later sections.

Let's focus on physical objects, the industrial artefacts. We give them identifiers, which generally stay fixed over time. The artefacts are characterised according to their qualities, form and function, material composition, and more.

Qualities of objects are generally allowed to change over time. With IDO, we follow a widely accepted practice of treating qualities as individuals. For instance, the temperature of an object is considered an object, as are the function and possible roles of the object. Clearly, such quality individuals only exist given that the bearer of the qualities exists as well – their existence is dependent on the objects that possess the qualities.

This means adopting a basic distinction between *independent*, quality-bearing objects, and *dependent* individuals, which carry the physical quantities, functions, and more, serving to capture the facts that need to be managed. In IDO, the class *Object* covers the independent objects, while dependent objects fall under *Dependent [entity]*.

[Ed. note: information objects are "generically dependent" in BFO ontologies, but currently placed under Object.]

[Ed. note: the term "dependent" replaces the former "aspect" to avoid confusion with the use of "aspect" in ISO/IEC 81346. Note, on the other hand, that annotations in the OM and SSN ontologies use the term "aspect" in accounting for qualities.]

Object 3D, independent, identified and constant over time

Dependent [entity] dependent, changing over time

hasQuality the basic relation from objects to qualities

7.2 Processes and Temporal Regions

Any industrial artefact will have its purpose in activities, whether this is production, manipulation of materials, or static structural support. The industrial artefacts and products *participate* in activities. Taking place over time, an industrial process is considered "four-dimensional". We work from an informal framework of relative speeds and absolute sizes, akin to "classical mechanics".

We may abstract away from any particular process, to talk about periods or instants of time themselves, disregarding any participation. With IDO, these are considered four-dimensional as well.

Being based in OWL, IDO is precluded from capturing any rich logic of time in its semantics. The vocabulary of four-dimensional entities however still allows us to capture a broad category of temporal relations: how processes and periods are ordered in succession, overlapping in time, and so forth.

In IDO, following the terminology of ISO 15926-2, the class *Activity* covers any 4D entity, with the following specialisations.

[TODO Comments to be worked into the text: (1) The everyday use of the term *event* does not assume that an event is instantaneous - e.g., sports event, charity event, workshop. The use of "event" in IDO is "technical", without an ambition to fit all common uses, as is also common in the literature. (2) The tentatively added subproperty lis:hasTemporalExtent has lis:Activity as a domain, and lis:TemporalRegion as a range. This is a different pattern to lis:hasSpatialExtent that has lis:Site (rather than lis:SpatialLocation) as a range. Temporal and spatial patterns should be similar, and if not, this should be explained and documented.]

Activity 4D, extending over time

Event Instantaneous occurrence, such as the end of a process

PeriodInTime A period, to which processes may be related

hasParticipant the basic relation from activities to objects

occursRelativeTo the basic relation of temporal order

7.3 3D versus 4D

Importantly, 4D objects do not have qualities in the way that 3D objects do. With IDO, any change effected by a process needs to be accounted for in terms of changes to the qualities of participants.

To represent quality change during a process, IDO provides the class *activity profile.* [Ed. note: this has been added to IDO, building on B. Smith's work on *process profiles*.] Any process under a common description such as "heating", "transport", or "oscillation" involves a multitude of qualities varying at the same time. However, we often need to refer to the changes of specific qualities – of temperature, location, or frequency – effected by the process. For this, we stipulate the existence of a part of the process that only consists in changes of the quality in question. The vocabulary is not quite settled – we may want to say that an activity profile is a "projection".

For example, to represent that a participant object "maintains a temperature of 50 degrees Celsius" during a process, we refer to the temperature quality of the object, and apply an appropriate aggregate notion, such as averaging, to values that quantify the quality with a unit of measure. The (dependent individual) changing quality, and aggregation, allows us a basis for appropriate classification of the process as "being performed at 50 degrees Celsius".

ProcessProjection part of a 4D object, limited to the variation of a specified quality of a participant

projectionDatum the basic relation from processes to data that quantify qualities of participants

7.4 Information

With IDO, the contents of documents or database records belong to the class of information objects. These are dependent objects, in the sense that they must be given *some* physical form to exist, whether as a paper printout or configuration of electrons in computer memory. They are "generically" dependent, as there may be multiple copies.

Information objects serve to describe objects, whether by naming them, quantifying measured qualities, or specifying the outcome of planned work. The core notion is "aboutness", and the usefulness of information generally involves comparison, as in specified information versus observations.

Following BFO, a physical representation of an (immaterial) information object is called a "concretization".

InformationObject immaterial, generically dependent, identified, and constant over time

isAbout the basic relation from information objects to the objects they characterise

concretizes the relation from an information object to a physical representation

7.5 Parthood (lis:hasPart)

Facts about part/whole relationships are captured by the hasPart relation, which has several distinct variants (subrelations). We need to represent the physical break-down structure of an industrial facility, or the assembly structure of equipment; processes and sub-processes; locations and sublocations; the chapters and sections of technical documentation; the functional parts of a system.

A proper account of parthood will often involve capturing complex interactions between entities that belong to very different domains, such as the 3D physical objects, the 4D processes in which they participate, and the immaterial domain of information.

For physical objects, the class Feature is of particular interest, covering parts that are inherent to the objects, yet not dependent in the way that qualities are. The primary case is that of surfaces, which need to be managed for correct interfaces between artefacts.

IDO provides a set of basic relations to support the different kinds of parthood.

hasPart the basic relation from whole to part

hasActivityPart process to sub-process

hasArrangedPart physical whole to the parts that make it up

hasFunctionalPart system to sub-system

hasSubLocation location to sub-location

7.6 Connections (lis:connectedTo)

The network of connections between parts of an industrial plant, or any parts of a technical system, is called a topology. This is captured by the 'connected to' relation.

When fasteners are used to affix a component, or an electrical connection is made between plug and socket, we have a *direct* connection between the objects. Indirect connections are also of interest, such as that between a battery and a light bulb; these are seen as derived from direct connections.

In practical models, we introduce classes like 'bolted connection' or 'electrical connection', with appropriate constraints on the kinds of objects that may participate. These constraints will be expressed using the basic vocabulary of "connected to", together with classes of artefacts.

connectedTo the basic relation between connected parts

directlyConnectedTo between objects that connect without intermediaries

7.7 Locations (lis:locatedRelativeTo)

Two distinct notions of location are recognised in IDO: geometric spaces defined with reference to an origin in a coordinate system, and conventionally defined spaces determined by reference to objects.

For both, we recognise points, areas, and volumes. Note that the two distinct notions of location will generally apply at the same time. In a building, the "front office", determined by the walls of a room, will coincide with a cube in the building's 3D model.

[Ed. note: There's a gap here. The *conventional* identification of a site doesn't trivially reduce to identification in terms of "surrounding objects". Discuss how the "front office" can stay the same, even if it moves to a different room.]

SpatialLocation a location defined by geometrical coordinates

Site a location defined by objects as boundaries

7.8 Qualities, especially physical quantities (lis:hasPhysicalQuantity)

In industrial use cases, the qualities of interest are, for the most part, physical quantities. These are dependent objects (see above), capable of being named and related to quantified values with units of measure (i.e., data).

The approach to quantities in IDO follows an established pattern that we find in several different ontologies for sensor data, quantification, and units. A physical object has physical quantities; the quantities are observed or specified in various ways; the resulting data points have numeric values and units of measure.

IDO naturally supports a taxonomy of physical quantities, structured according to SI dimensions, with an accompanying vocabulary of units of measure. These can be used in defining classes of industrial artefacts, with ranges of physical attributes represented as proper semantic restrictions. By classification of the quantifying data, distinctions can be made between measured, stipulated, or specified values.

IDO is open to introducing *relational* qualities, such as the pressure "delta" between two points in a model, with physical quantities that essentially depend on more than one physical object. [Ed. note: add reference to BFO/B. Smith] (As of Q2 2023, the IDO vocabulary doesn't provide any resources specifically for modelling relational qualities, but such could be added in RDL.)

The vocabulary for physical quantities includes

PhysicalQuantity the class of physical quantities

ScalarQuantityDatum the class of data representing quantity magnitudes

UnitOfMeasure The class of units

hasPhysicalQuantity relation from physical object to a (dependent) quantity

qualityQuantifiedAs relation from a quantity to a datum

datumUOM relation from a datum to a unit-of-measure individual

datumValue relation from a datum to a literal number

7.9 Disposition, Capability, and Function (lis:realizedIn)

The IDO approach to dispositions is adopted from Basic Formal Ontology (BFO) and related works.

Each artefact in an industrial plant is there to serve a function (a purpose), and the function will be represented as a dependent individual. An artefact may also have capabilities, secondary to the primary purpose; for instance, where a piece of heavy equipment also has a useful stabilisation effect. We also recognise the weaker notion of dispositions, which may be unwanted, such as the disposition of metals to rust, or of vessels to leak.

Every function is a capability, and every capability is a disposition – this hierarchy allows for a suitable degree of distinction between types of behaviour, from the overall level of plant systems down to individual pieces of equipment.

Following BFO, we characterise a disposition (resp., capability; function) of an object by the kinds of *activities* that count as its *realisations*. For example, the function of a pump is realised when it participates in a pumping activity, providing pressure increase within the intended range.

Of interest in risk and maintenance scenarios, known ways that artefacts can malfunction may also be represented by way of dispositions. Vocabulary for this purpose is not included in IDO itself, but may be added in RDL extensions.

A disposition may or may not be realised during the object's lifetime, dependent on conditions. A piece of metal kept dry will not rust; a fire extinguisher may never need to be used.

Disposition what an artefact tends to do

Capability a favourable disposition that is not a primary purpose [Ed. note: review B. Smith's presentation]

Function a capability that is the reason for the object's existence or presence

7.10 Activity and Participation (lis:participantIn)

Participation in activities can take many forms, including as agent, material, or output. IDO leaves the task of distinguishing specific kinds of participation to an RDL (a domain-specific ontology that extends IDO). [Ed. note: but we could consider adding main categories to IDO itself.]

For example, the systems modelling framework IDEF (initially by US DoD) provides a source of main participation types: Input, Output, Control, and Mechanism/Resources. A vocabulary of kinds of participation needs to be chosen according to the needs of a use case.

For an activity to qualify as the realisation of a function of an object, the participating object will typically need to participate as an *agent*; as in, the pump needs to be pumping. This is however not without exception; depending on the use case, the type of participation of an injected chemical catalyst, or fuel for a motor, may be more appropriately described in different terms.

Due to the expressive limits of OWL, an IDO based model will in general be incomplete in its semantic representation of the interplay between artefact and activity. [Ed. note: DLs are restricted to tree structures; lack "diamond" structures. A full discussion is beyond scope for the IDO document. Provide references to literature.] This calls for the implementation of rules, and accompanying reasoning capabilities, which are outside the scope of IDO itself.

7.11 Organisations and Interests (lis:hasInterest)

The class supports the representation of actors engaged in an industrial effort, including operators, vendors, service providers and public bodies. The hierarchy and set-up of an organisation may be characterised with part/whole relationships, using the hasPart relation, and the participation in activities using the participantIn relation.

IDO doesn't define a class for personnel, but this may be added in an RDL, as a subclass of Object.

There is a range of intentional relationships between actors and industrial entities that benefit from semantic representation, such as the assessment of a procedure, the review and acceptance in a handover, or the approval and publication of an industry standard by a standards body. IDO offers *interestOf* as a covering relation for these needs. [Ed. note: clarify to what extent this could be, and not be, covered by variants of participation.]

Organization the class of groups of people with common agency and intentions

interestOf basic relation for attitudes of an organisation towards activities

7.12 Roles (lis:hasRole)

TODO

Like dispositions, roles are described by way of the activities that count as realisations.

7.13 Prescriptive and Installed objects (lis:installedAs)

A primary aim for IDO is to support some degree of automated checking of requirements against (proposed or built) solutions.

The core of a typical plant model expressed with IDO will have a character of *prescription*. It will describe the plant as a graph of individuals, with break-down structures and connectivity; and individuals will primarily carry *constraints*, captured in semantic constraints on their classifiers. The common notion of *tag* is suitable as a mental model for prescription. For example, a pump individual (as a tag) has constraints on performance, size, and power use.

The semantic model also needs to represent the plant *as built*, with named individuals in the model that represent "concrete", installed artefacts. During the life of a plant, some of the individuals will need to be replaced. Checking requirements is then essential.

For the representation task, IDO provides the relation 'installed as': where x is an individual in the designed plant model (the tag), various specimens $y1, y2, \ldots$ will be installed over time. Each specimen needs to be checked for compatibility with the requirements on the tag.

Given that constraints are captured uniformly, the suitability of a candidate item to serve in the position of a prescriptive individual (a tag) can be carried out with precision. In brief, we check that the constraints on a prescriptive individual are consistent with the constraints on a proposed installed individual. A prerequisite is that the plant model specification and the description of built products are given in the same language. If properly implemented, the automated reasoning capabilities of the OWL language can handle a broad range of industrially relevant verification tasks.

The “installed as” relation as given in IDO is purposely made semantically weak, to allow for a prescriptive individual (tag) to be related to an arbitrary number of installed artefacts over the lifetime of the plant (and regardless of whether they satisfy the constraints). The algorithm for checking each installed artefact against the tag requirement will be described in detail elsewhere in this standard.

[TODO update with recent improvements, in particular classes *Prescriptive* and *Actual*.]

installedAs relation between a prescriptive individual and concrete installed individuals

7.14 Ontology Evolution (lis:originatesFrom)

As an industrial facility is modified over its lifetime, the semantic asset model needs to be updated accordingly.

Because OWL lacks temporal expressions, it’s not practical to capture the history of changes of an IDO plant model in a single ontology. It is instead recommended to maintain a series of historical snapshots of the plant model. Comparisons across time can be made using queries using a language like SPARQL.

The evolution of a plant may mean that the role initially done of a single piece of equipment is later divided between two, or vice versa. To track such changes, IDO provides a set of annotation properties. These serve as pointers “looking back” from an individual in a version of the asset model, to one or more individuals in earlier versions. By default, these annotation properties apply to *prescriptive* individuals (tags).

originatesFrom the basic relation

mergedFrom for an entity that replaces two or more earlier entities

splitFrom for separate entities that replace what was previously a single entity

transformedFrom for an entity that replaces an earlier entity with a significantly different character

7.15 Metadata

Being built on the Resource Description Framework (RDF) language, an IDO ontology can make full use of the metadata facilities of the Semantic Web.

Annotations on individuals, classes, and relations, as well as on ontologies themselves, can employ vocabularies such as the widely used *schema.org*, *Dublin Core*, *SKOS*, and others.

For metadata on the ontology resources in IDO itself, we follow the norm developed by the Industrial Ontology Foundry (IOF), as captured in the *IOF Annotation Vocabulary*. This covers metadata such as proper naming, definitions and explanatory text, examples, provenance, and more.

Annotations of complex semantic constructs are also practical. For example, restrictions on physical quantities (represented in OWL axioms on requisite classes) can be provided with metadata.

7.16 Second-order notions (SKOS)

The OWL language provides the capability of classifying not only plant individuals, but also classes and relations, by the (strictly non-semantic) device of “punning”. This is useful in the management of vocabularies.

IDO recommends the use of the Simple Knowledge Organization System (SKOS) for classifying, grouping, and relating vocabulary items. The SKOS vocabulary provides basic classifiers relations on both individual vocabulary items and structured vocabulary collections, including the following.

Concept the general class of terms

Concept Scheme a structured vocabulary

is in semantic relation with top relation for a “thesaurus” vocabulary

Annex A
(informative)

Use cases

A.1 Pump with Firmware

A.1.1 Specific Challenge

This use case shows how the IDO ontology is used to develop an application ontology and associated knowledge graph for an engineering product. For illustrative purposes we use a pump manufacturing company (ACME) which designs, manufactures, sells and provides lifecycle support services for pumps. Their pumps are sold to thousands of customers and installed in many locations.

ACME sells a centrifugal cooling pump, type *E-78-130-F-M-270* (material master) running firmware *SupraCool2000_FW_2_23_345* to the MilkyMasterHaderslev process facility (a dairy process plant). ACME also has a department called ACME_DigitalDairy. ACME_DigitalDairy sells a predictive maintenance monitoring service to MilkyMasterHaderslev.

The cooling pump has a controller. The controller contains firmware. This firmware holds software which is updated over time and the pump organisation needs to track the version of software installed on each pump. This includes information such as the internet addresses and settings used in IoT communication, as described using the W3C IoT Web of Things (WOT) Thing Description (TD) files.

The ACME internal digital user manual describes how ACME_DigitalDairy can use the pump as an IoT Device to request liquid fluid temperature as a "Web of Thing (WoT) thing description" JSON file. The data comes from a temperature sensor installed in the pump housing. Data is captured and stored in the pump's control system and can be transmitted directly to ACME_DigitalDairy using IoT by means of a Modbus protocol. This temperature reading is needed by ACME_DigitalDairy for the predictive monitoring service.

The semantic asset model below describes the physical pump and its control system, the sensing system installed in the pump, the firmware and software, and the JSON file. The semantic model is used by all business groups in ACME and holds enough information to enable the digital business of ACME. Using the IDO model as a reference and additional classes, relations and instances relating to the ACME use case, the resulting model supports tracking of the data as it is moved from the sensor through various transformations to the IoT communication, capturing units and values used in the transformations, and the versions of software used to effect these transformations and communications. All classes, relations and instances relating to the use case are given the prefix *ex:* while classes and relations relating to IDO use the prefix *lis:*.

A.1.2 Competency Questions

Competency questions represent a good practice approach to defining requirements for ontology development. Answering competency questions should add value for stakeholders of the ontology and the ability to address competency questions should be a meaningful test of the ontology. Once the ontology is developed, the competency questions can be implemented e.g. in the form of SPARQL queries that use the ontology as a schema.

This example is designed to illustrate the use of IDO to model an IoT-enabled centrifugal pump to address the competency questions below.

Identify the calibration range of the temperature sensor installed in the pump.

Check whether the unit of measure and the measurement range of the temperature sensor are consistent with IoT WoT TD specification.

Provide a liquid temperature reading to the user (ACME-DigitalDairy)

In order to make this example tractable and easy to understand the following are considered out of scope:

The components not directly involved in the temperature sensing system.

Make and model of the sensor and its associated hardware.

A.1.3 Equipment Description

A schematic of a pump with controller and firmware elements is shown in Figure A.1. Elements are described using numbers as follows.

1. pump housing
2. impeller
3. shaft
4. shaft seal
5. motor
6. temperature sensor (installed inside the housing)
7. electronics controller housing
8. touch screen display for the sensor data and user input
9. electronics
10. electric power
11. connectivity
12. motor and pump nameplate.

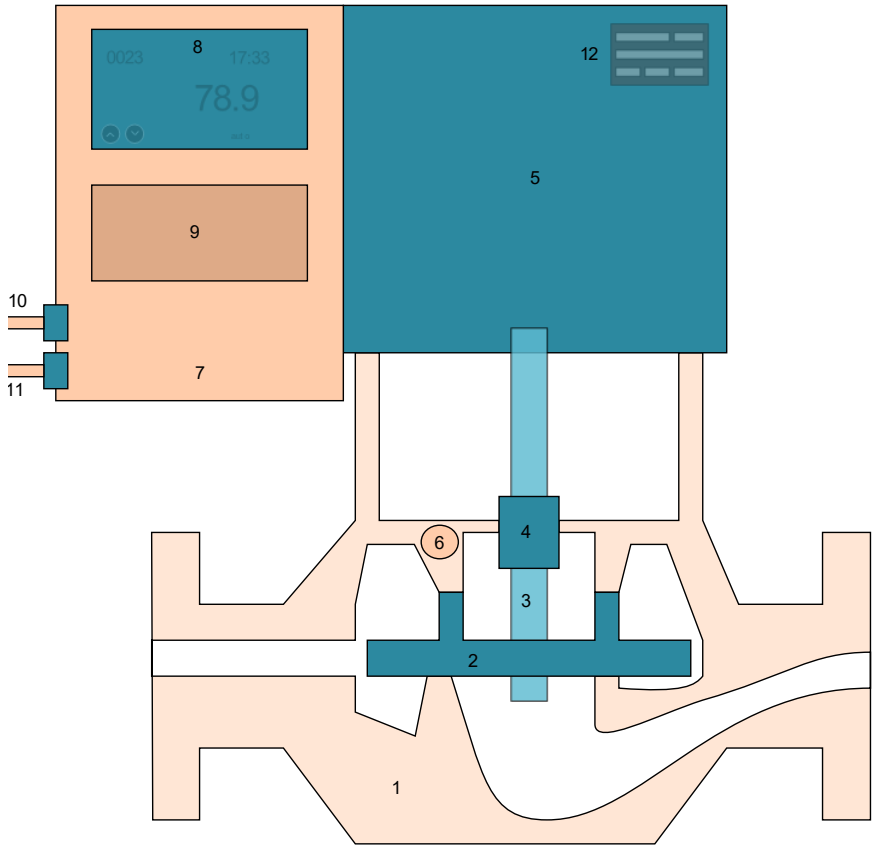


Figure A.1 — Centrifugal pump schematic with components

A picture of a real pump which is the basis of this schematic is shown in Figure A.2.

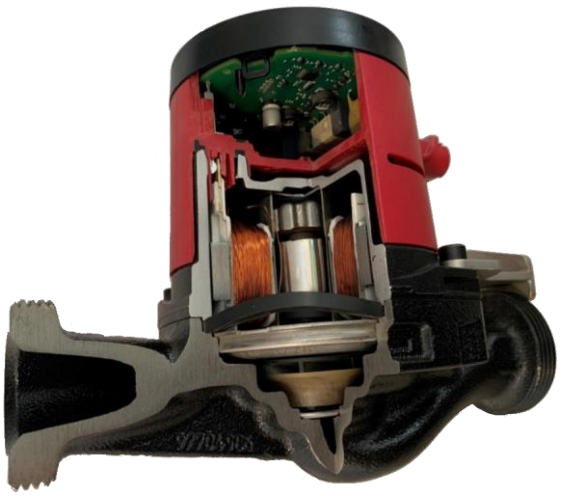


Figure A.2 — Cut-away view of the internals of the pump used in the schematic in Figure 9.2.

A.1.4 Modelling the pump

At the top level in the semantic model there are two physical objects (individuals) that enable the IoT call for reporting of a "Liquid Temperature". These are 1) the ACME pump (with IRI *ex:5000.02.01.PU001.Pump*) which includes both hardware and software elements, and 2) the pumped fluid (with IRI *ex:PU001.PumpedFluid*) that has the fluid temperature of interest to ACME DigitalDairy.

IDO allows us to model that 1) the pumped fluid is contained in the pump, and 2) the temperature (with IRI *ex:PU001.FluidTemperature*) is a quality of the pumped fluid as shown in figure A.3 using the IDO object properties *lis:contains* and *lis:hasPhysicalQuantity*.

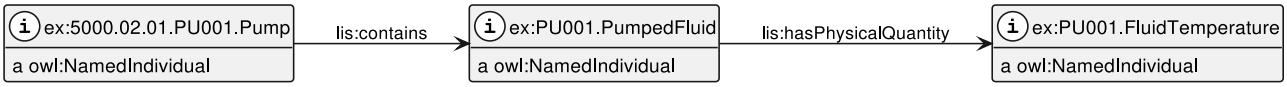


Figure A.3 — Pump, fluid, and fluid temperature

We can also model what *kind* of thing the pump is. The pump is an instance of the class *ex:E-78-130-F-M-270*. This is a specific leaf in a pump taxonomy that is anchored in the IDO class *lis:PhysicalArtefact* as seen in figure A.4. In this example the ACME company, rather than make *ex:E-78-130-F-M-270* a subclass of *lis:PhysicalArtefact*, might use an equipment hierarchy, in this case the class *ex:Pump* and its subclass *ex:CoolingPump* to which *ex:E-78-130-F-M-270* can be mapped. Domain specific equipment taxonomies from the Standards literature such as IEC 81346-2, ISO 15926-4 or the IEC's Common Data Dictionary (CDD) could also be used.

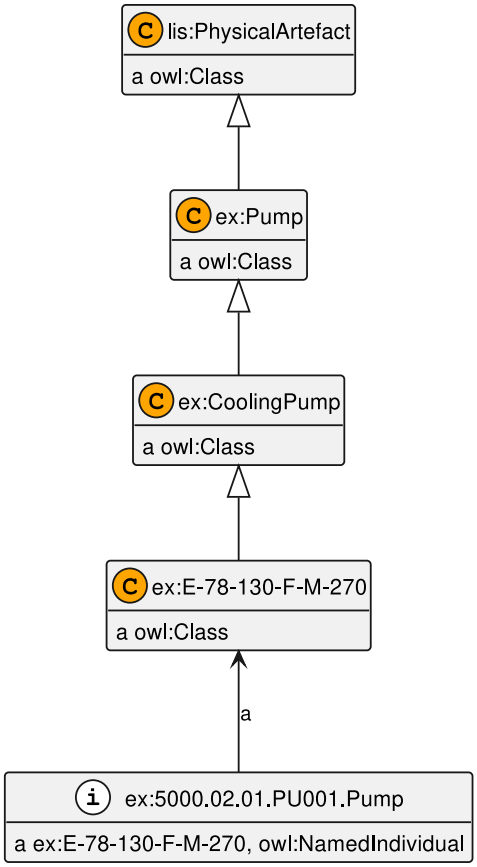


Figure A.4 — Taxonomy from pump type E-78-130-F-M-270 to IDO class lis:PhysicalArtefact

To answer the competency questions, we need more details in the model, more than it is useful to plot in one figure. In the following we expand the ACME application model into different submodels. Collectively these submodels hold the information necessary for the full data model and associated reasoning. Each submodel is selected to illustrate a specific concept or approach described in the text. To avoid cluttering the figures, superclasses and some relations may not be shown in a specific figure. Instances representing objects of interest are added to produce a knowledge graph. The full model, serialized in RDF is available at listing <xxxxx> [TODO include listing].

In Figure A.5 we model the pump components “pump housing”, “temperature sensor” and “electronic controller”, all classes anchored in an ACME taxonomy under *lis:PhysicalArtefact* (not shown). Note that the IDO object property *lis:connectedTo* is used to represent that the temperature sensor is attached to the pump housing (measuring the pump housing temperature). Below we will model how the liquid temperature is estimated from the pump housing temperature.

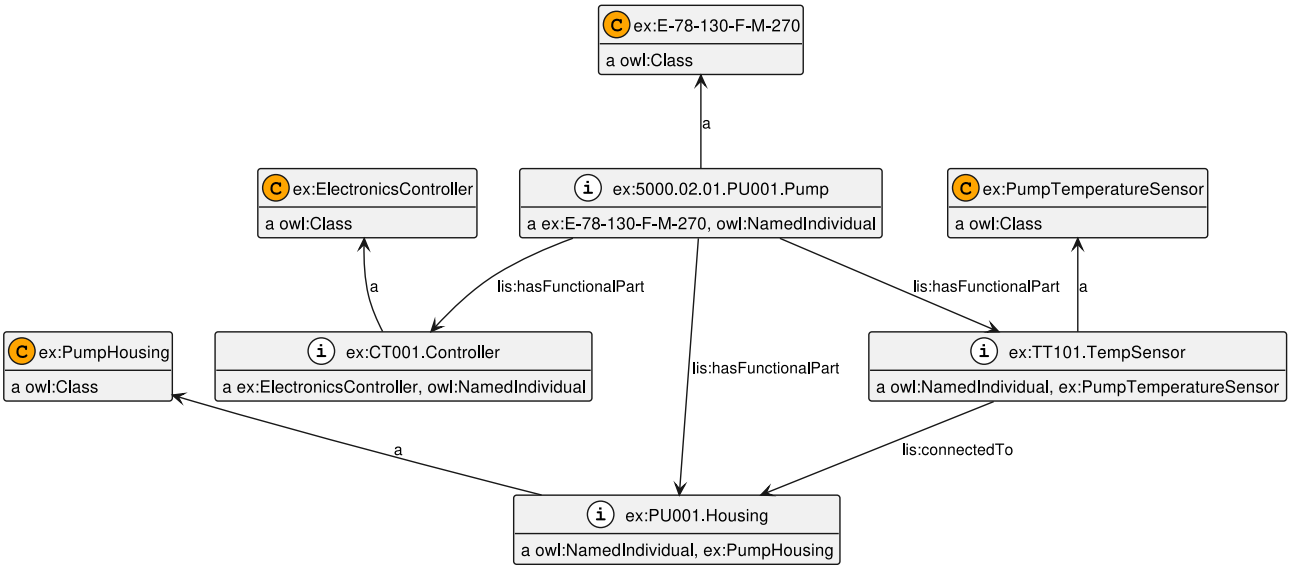


Figure A.5 — 5000.02.01.Pu001.Pump and functional parts

A.1.5 Temperature Observation

The sensor is assumed to be a fully cyber-physical system (not modelled in detail) with its own hardware and software, sourced from another vendor than the pump manufacturer.

The pump's controller has as a component installed firmware *ex:CT001.ConstrolScheduler.Firmware* that on a regular schedule initiates a temperature sensing activity. This activity has, as a part, an activity profile *TT101Temp.Observation* as shown in Figure A.6. This is a subactivity of performing a single measurement using the Sensor. *ex:TT101.TempSensingActivity* also has other (not shown in Figure 9.6 but illustrated in Figure A.9) activity parts that are responsible for obtained data being stored in a register on the controller for later retrieval etc., during which some controller firmware down-samples the 14 bit result, the sensor provides to an 8 bit format that fits the register, among other processes. Included is the calibration of the Sensor modelled as a quality of the Sensor *ex:TT101.TempSensor*. In IDO, an equipment 'participates in' activities, so we have modelled *sosa:madeBySensor* as a subproperty of the IDO object property *lis:hasParticipant*.

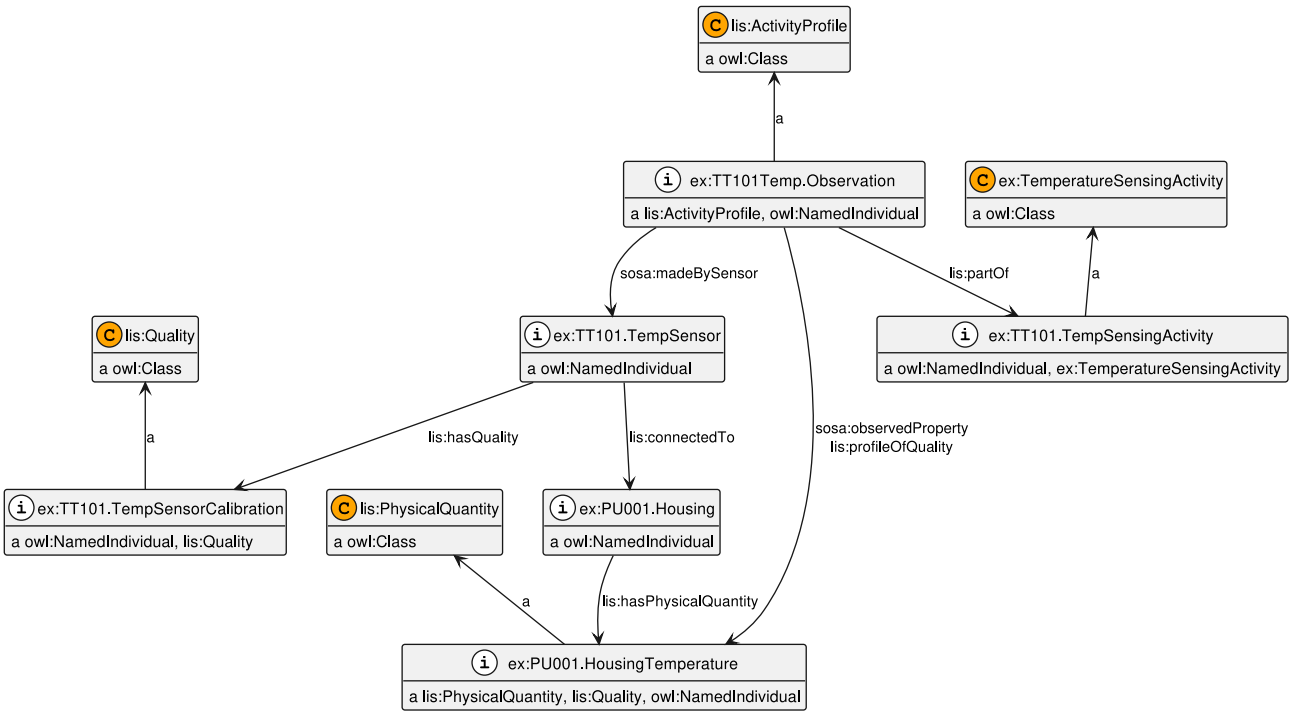


Figure A.6 — Classes, relationships and instances to model a temperature observation

A.1.6 Firmware and Software and their instantiation

We have identified a pattern around firmware that is installed on electronic devices (firmware). Developers have developed an software artefact *ex:CT001.ControlScheduler.Software*. This is installed on the pump into the electronic memory of the pump controller, we identify this concretization of the software as installed software as *ex:CT001.ControlScheduler.Firmware*. The software is developed with a function in mind. We model this as a relation between the installed firmware *ex:CT001.ControlScheduler.Firmware* and the function *ex:CT001.ControlScheduler.Function*. This pattern is shown in Figure A.7.

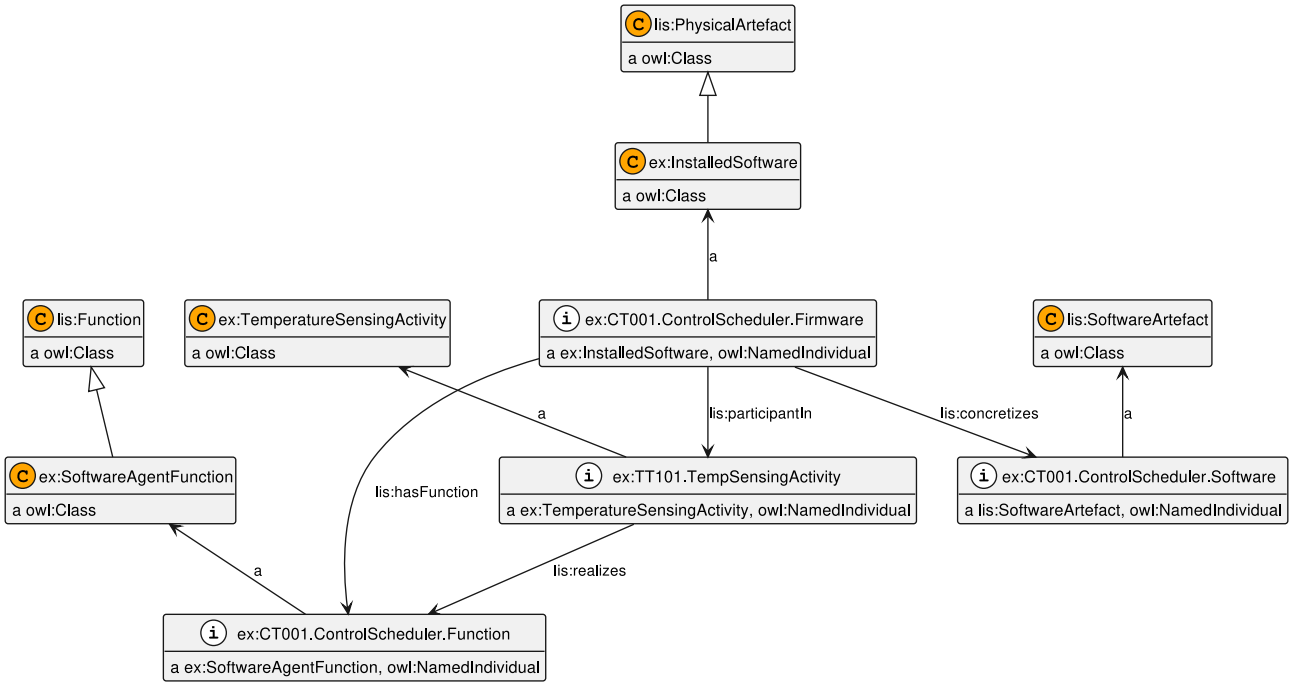


Figure A.7 — Pump controller functions involved in temperature sensing

A.1.7 Temperature measurement and downsampling activity

The Temperature measurement activity profile *ex:TT101Temp.Observation* produces the quality datum *ex:TT101.tempReading001*. The datum has a resolution that we model with the bespoke data property *ex:hasResolutionInBits*. The 14 bit temperature datum *ex:TT101.tempReading001* acts as input to a downsampling software process *ex:TT101.DownSamplingActivity* that produces another temperature datum *ex:TT101.tempReading002* that still quantifies the housing temperature, but with a lower resolution than *ex:TT101.tempReading001*. Being in control and able to model these details will be key when managing a large portfolio of electronic pumps with different sensor types and different resolution choices implemented.

We simplify the diagram by modelling the downsampling activity as a subactivity of *ex:TT101.TempSensingActivity* which means we consider the firmware *ex:CT001.ControlScheduler.Firmware* to hold both the firmware for scheduling the sensing activity and also firmware for downsampling as parts. We model the information flow and processing the temperature datum in the pattern of *lis:Datum – lis:participantIn → lis:Activity – lis:representedIn → lis:Datum*, where we have introduced the property *ex:hasInput* to model a Software activity input. This pattern is shown in Figure A.8.

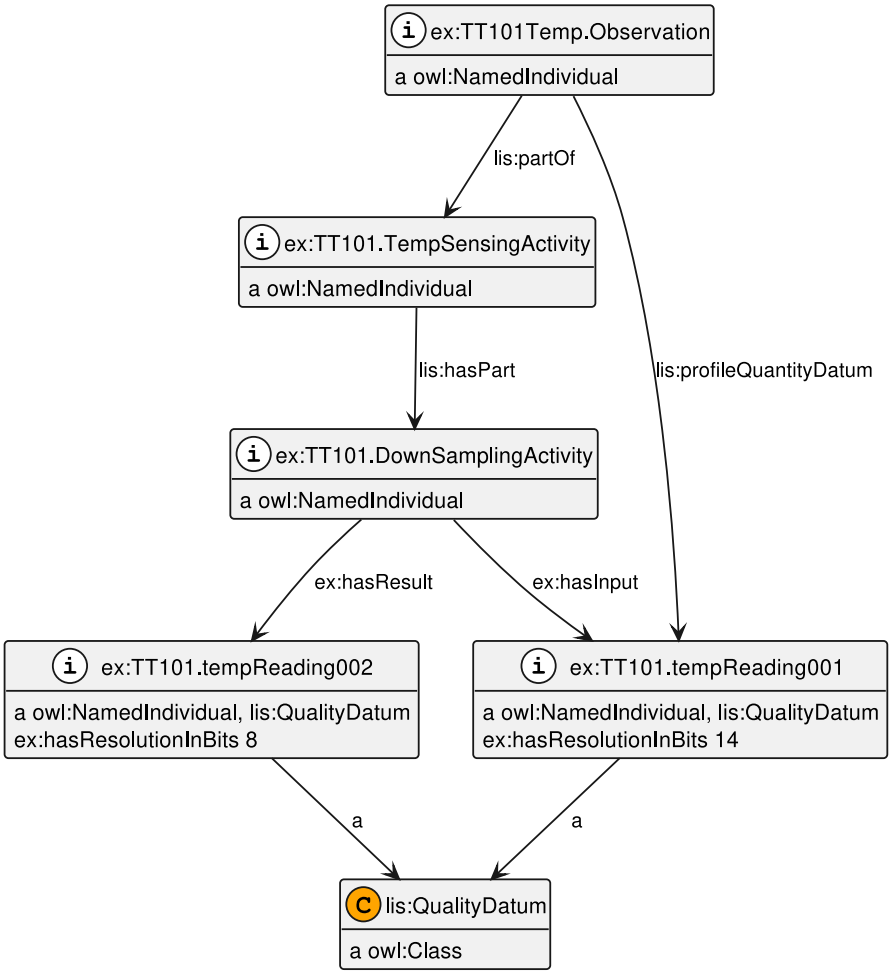


Figure A.8 — Downsampling of the temperature reading

A.1.8 The IoT Call aligned with Web OF Things TD

The pump can deliver an estimation of the liquid temperature to some other connected system (another controller at the same site, or to the cloud) and in this example it is designed to do this in the "Modbus" protocol. The objective of this modelling exercise is to maintain information about the lineage of such IoT temperature data points, and to maintain information to be distributed to stakeholders in a digital manner using the "Web Of Things" framework in a "Thing Description" (TD) JSON file as seen in listing A.10.

In the above description, there is no mention of the liquid temperature, only the pump housing temperature. The lineage is that the liquid temperature estimation datum *ex:PU001.FluidTemperatureEstimation_TimeStampedIndex54678* is estimated in a software process activity *ex:CT001.IoTServerProcess001* of which the pump housing temperature is an input. This software process activity *ex:CT001.IoTServerProcess001* also delivers the liquid temperature estimation datum to the right output port adhering to the Modbus protocol. This is shown in Figure A.9.

The sensor calibration *ex:TT101SensorCalibration* together with the resolution in bits of *ex:TT101.tempReading002* and the estimation algorithm some developer has coded into *ex:CT001.HandleTempIoTCallAndTempEstimation*. Software are all necessary prerequisites for calculation of both the liquid temperature estimate and the WoT *multipleOf* parameter that conveys how many decimal places of the IoT data you should trust.

The software is versioned using semantic version with major, minor and patch counters.

Another WoT parameter that the model holds is the server API address "1/40000?quantity=1" of the liquid temperature IoT functionality.

Now the model can report where to connect to obtain liquid temperate data (API address "1/40000?quantity=1"), That this is Estimated data, what software artefacts (and by extension what

semantic versions) were involved the lineage of the liquid temperature estimate, what sensor with what resolution this estimation is based on, etc.

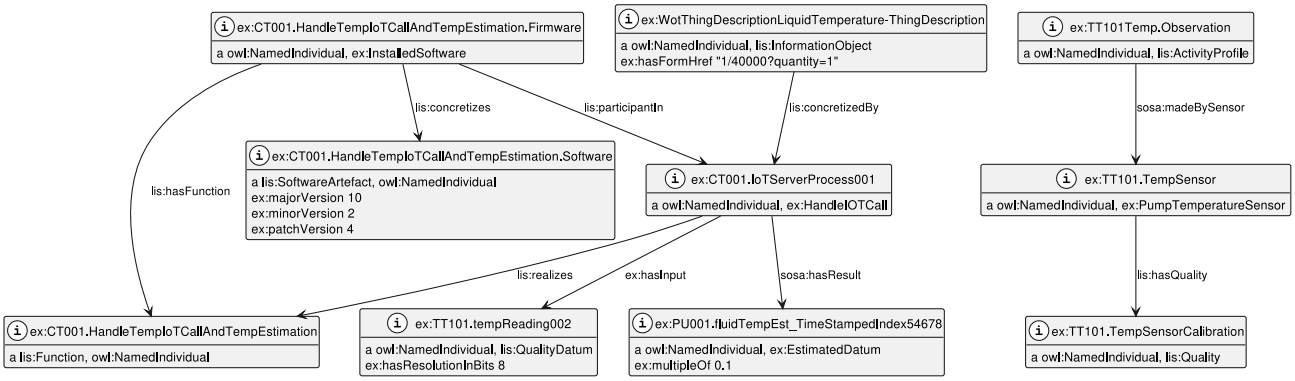


Figure A.9 — The IoT call handling process

A.1.9 The resolution of the temperature estimate

A key challenge this use case is modelled to solve, is the management and calculation of information about measurement uncertainty of sensors and how the storage space of data registers (8 bit versus 14 bit). The sensor has 14 bit resolution on the range of 0–95 °C (unit of measure not modelled, but this is shown in other PCA usecases). The 14 bits can represent 65536 different values (with perfect calibration this would correspond to a resolution of approx. 0.0014 °C). However, the chosen memory on the pump electronics only handles 8 bit resolution, corresponding to 256 different values (i.e. storing a temperature range of 0–95 °C in steps of approx. 0.4 °C). The best possible output of the liquid temperature estimation process will therefore also have a resolution of 0.4 °C.

A.1.10 Web of Things Example

A feature of this use case is to demonstrate how the instance data captured in the model is used in the WoT communications for a pump (JSON file). An example of a WOT compliant protocol is shown in Listing A.10. The ACME corporation owns a pump of type *ex:E-78-130-F-M-270* which has a unique identifier *assets:3ea24c16-afc2-4d61-af6a-2964d76f160a*, identified using an "id" key-value pair.

The aim is to expose the *acme:LIQUID_TEMP* as a "property affordance" in the terminology of WoT. In the property JSON object *acme:LIQUID_TEMP* various meta data are exposed to IoT developers, as seen in the details below.

The resolution of the Sensor, the controller constraint of 8 bit register for the data, and the temperature range that the sensor is calibrated to operate within, deterministically puts a lower bound on the resolution of the IoT datum presented through the Modbus protocol of "multipleOf 0.4". The company ACME, based on these insights, decides to report a precision in the public WoT TD JSON of 1 °C, adding a small safety margin.

```
{
  "@context": [
    "https://www.w3.org/2019/wot/td/v1",
    {
      "acme": "https://www.acme.com/linkeddataserver/iotprotocol#",
    "assets": "https://www.acme.com/linkeddataserver/assets#",
    "unit": "https://www.qudt.com/unit#",
    "modbus": "https://www.w3.org/2019/wot/modbus#",
    "lis": "http://rds.posccaesar.org/ontology/lis14/rdl/",
    "ex:": "http://example.org/pumpwithfirmware"
    }
  ],
  "id": "assets:3ea24c16-afc2-4d61-af6a-2964d76f160a",
  "@type": [
    "Thing",
    "ex:E-78-130-F-M-270"
  ],
  "title": "E-78-130-F-M-270",
  "properties": {
```

```
    "acme:LIQUID_TEMP": {
      "type": "number",
      "readOnly": true,
      "description": "Liquid Temperature",
      "lis:PhysicalQualtity": "ex:Temperature",
      "unit": "unit:DEG_C",
    "multipleOf": 1,
      "forms": [
        {
    "href": "modbus+tcp://127.0.0.1:60000/1/40001?quantity=1",
    "op": [
      "readproperty"
    ],
    "modbus:entity": "HoldingRegister"
    }
      ]
    }
  "acme:VOLUMETRIC_FLOW": {
      "type": "number",
      "readOnly": true,
      "description": "Volumetric Flow",
      "lis:PhysicalQualtity": "ex:VolumetricFlow",
      "unit": "unit:M3-PER-SEC",
    "multipleOf": 0.01,
      "forms": [
        {
    "href": "modbus+tcp://127.0.0.1:60000/1/40001?quantity=2",
    "op": [
      "readproperty"
    ],
    "modbus:entity": "HoldingRegister"
    }
      ]
    }
  }
}
```

Figure A.10 — Web of Things protocol

A.2 Candidates for further use cases

- Single-line diagram
- Alloys
- RDS for wind power
- FMEA use case
- Asset model

Annex B
(informative)

Extending IDO with special purpose ontology standards

IDO offers a generic vocabulary for industrial asset representation, which can be enhanced with established specialized vocabularies.

[TODO. Compare with ISO/IEC 21823-3:21 (Interoperability for IoT systems, Part 3: Semantic interoperability), informative Annex E "Related existing ontologies", where SSN, SAREF, and other ontologies and ontology-like languages are described. Apparently, no mappings or guidelines to specific integrations are given in the annex.]

The paper *A Framework Uniting Ontology-Based Geodata Integration and Geovisual Analytics* (Ding et al., 2020) describes an integration of three widely used ontologies, for Ontology Based Data Integration and geovisual analytics:

GeoSPARQL, a standard by the Open Geospatial Consortium, 2012

Semantic Sensor Network (SSN), a W3C Recommendation, 2017

Time Ontology in OWL (OWL Time), a W3C Candidate Recommendation, 2022

The following sections describe mappings to allow these ontologies to be combined with IDO.

Some ontologies already provide mappings to other generic ontologies. For the SSN, the section "Vertical Segmentation" of the ontology documentation provides mappings to DOLCE and other ontologies. The mappings provided below are designed to follow the same approach.

B.1 Alignment with the Semantic Sensor Network ontology

See https://www.w3.org/TR/vocab-ssn/

B.1.1 Namespaces

[TODO] There are two namespaces in use in SSN, ssn: and sosa:.

```
Prefix: ssn: <http://www.w3.org/ns/ssn/>
Prefix: sosa: <http://www.w3.org/ns/sosa/>
```

B.1.2 Class Alignments

Mapping: Semantic Sensor Networks classes to IDO

SSN		IDO
Actuation	subclass of	ActivityProfile
Actuator	subclass of	PhysicalObject
FeatureOfInterest	subclass of	Object
ObservableProperty	subclass of	Quality
Observation	subclass of	ActivityProfile
Platform	subclass of	PhysicalObject> ,
Procedure	subclass of	InformationObject
Result	subclass of	QualityDatum
Sample	subclass of	QualityDatum
Sampler	subclass of	PhysicalObject

SSN		IDO
Sampling	subclass of	ActivityProfile
Sensor	subclass of	PhysicalObject

B.1.3 Property Alignments

Mapping: Semantic Sensor Networks properties to IDO

SSN		IDO
actsOnProperty	subproperty of	profileOfQuality
hasFeatureOfInterest	subproperty of	qualityOf
hasResult	subproperty of	profileQuantityDatum
hasSample	subproperty of	qualityQuantifiedAs
hosts	subproperty of	hasFunctionalPart
isActedOnBy	subproperty of	qualityProfiledIn
isFeatureOfInterestOf	subproperty of	hasQuality
isHostedBy	subproperty of	functionalPartOf
isResultOf	subproperty of	quantityDatumOfProfile
isSampleOf	subproperty of	quantityDatumOfProfile
madeActuation	subproperty of	activeParticipantIn
madeByActuator	subproperty of	hasActiveParticipant
madeBySampler	subproperty of	hasActiveParticipant
madeBySensor	subproperty of	hasActiveParticipant
madeObservation	subproperty of	activeParticipantIn
madeSampling	subproperty of	activeParticipantIn
observedProperty	subproperty of	profileOfQuality
phenomenonTime	subproperty of	hasTemporalExtent
resultTime	subproperty of	timestamp

B.2 Alignment with the GeoSPARQL ontology

[TODO]

See https://www.ogc.org/standard/geosparql/.

The current version of the GeoSPARQL standard, version 1.0, was released in 2012. Version 1.1, is currently (Q2 2023) in progress and expected to replace the current version by the time the IDO is issued as a standard.

B.2.1 Namespaces

The following namespaces and prefixes are introduced with GeoSPARQL.

```
Prefix: geo: http://www.opengis.net/ont/geosparql#
Prefix: geof: http://www.opengis.net/def/function/geosparql/
Prefix: geor:  http://www.opengis.net/def/rule/geosparql/
Prefix: gml:  http://www.opengis.net/ont/gml#
Prefix: ogc:  http://www.opengis.net/
Prefix: sf:  http://www.opengis.net/ont/sf#
```

B.2.2 Class Alignments

B.2.3 Property Alignments

B.3 Alignment with the Time Ontology in OWL

[TODO]

See https://www.w3.org/TR/owl-time/.

B.3.1 Namespaces

The following namespace and prefix is introduced with OWL Time.

```
Prefix: time: http://www.w3.org/2006/time#
```

B.3.2 Class Alignments

B.3.3 Property Alignments

Annex C
(informative)

Supplementary resources

This section is intended for extensions – material that is part of the standard, but still optional. The ontology should be perfectly usable with these.

C.1 The use of datatypes

C.1.1 Floating-point numbers

It will be beneficial if we can recommend a default datatype for floating-point numbers.

xsd:float is an attractive choice, for allowing values of positive and negative infinity. Quoting Datatype-Rescue from W3C (2007, last update 2009):

The *value space* of float contains a copy distinct from owl:real and xsd:double of the non-zero numbers $m \times 2^e$, where m is an integer whose absolute value is less than 2^{24}, and e is an integer between -149 and 104, inclusive. In addition to these values, the *value space* of float also contains the following *special values*: positiveZero, negativeZero, positiveInfinity, negativeInfinity, and notANumber.

C.2 Rules in SWRL

If the OWL ontology is augmented with SWRL rules, we can obtain useful bridging relationships. These are suitable for use in property chains, enabling valuable inferences.

Example, for physical quantities (taken from CFIHOS presentation 09-2022).

```
Rule:
      hasPhysicalQuantity( ?x, ?y ), Pressure( ?y ) -> hasPressure( ?x, ?y )
```

Example, for realization of dispositions or roles (for illustration only).

```
Rule:
      hasFunction( ?x, ?y ), ToolFunction( ?y ) -> hasToolFunction( ?x, ?y )
```

The latter allows for a property chain to ensure that a tool function is only realized in activities where the function-bearer participates as a tool. The following example is for illustration only; compare with the property chain lis:realizes o lis:dispositionOf -> lis:hasParticipant.

```
ObjectProperty: lis:hasParticipant_Tool
  SubPropertyChain:
    lis:realizes o lis:toolFunctionOf
```

C.3 Ontology patterns in OTTR

We have the option of elevating OTTR modelling patterns to a required status. That is, we may choose to make some of the patterns that would ordinarily be described in use cases, as informative only, into required patterns.

Bibliography

[1] B. Motik, P. Patel-Schneider, B. Parsia, OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition), World Wide Web Consortium, 2012. http://www.w3.org/TR/owl2-syntax/.

[2] R. Arp, B. Smith, Function, role and disposition in basic formal ontology, Nature Precedings. (2008). https://doi.org/10.1038/npre.2008.1941.1.

[3] R. Arp, B. Smith, A.D. Spear, Building ontologies with basic formal ontology, The MIT Press, 2015. https://mitpress.mit.edu/9780262527811/building-ontologies-with-basic-formal-ontology/.

[4] EN 17632-1, Building information modelling (BIM) - Semantic modelling and linking (SML) - Part 1: Generic modelling patterns, CEN/TC 422, Brussels, 2022. https://standards.cencenelec.eu/dyn/www/f?p=205:110:0::::FSP_PROJECT:67839&cs=13BE091B11208910B30E53F9215AFDE96.

[5] ISO 15926-2, Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 2: Data model, ISO/TC 184/SC 4, Geneva, CH, 2003. https://www.iso.org/standard/29557.html.

[6] ISO 15926-12, Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 12: Life-cycle integration ontology represented in Web Ontology Language (OWL), ISO/TC 184/SC 4, Geneva, CH, 2018. https://www.iso.org/standard/70695.html.

[7] ISO 19450, Automation systems and integration – Object-Process Methodology, ISO/TC 184/SC 5, Geneva, CH, 2015. https://www.iso.org/standard/62274.html.

[8] ISO/IEC 21823-3, Internet of Things (IoT) – Interoperability for IoT systems – Part 3: Semantic interoperability, ISO/IEC JTC 1/SC 41, Geneva, CH, 2021. https://www.iso.org/standard/83752.html.

[9] ISO/IEC 21838-1, Information technology – Top-level ontologies (TLO) – Part 1: Requirements, ISO/IEC JTC 1/SC 32, Geneva, CH, 2021. https://www.iso.org/standard/71954.html.

[10] ISO/IEC 21838-2, Information technology – Top-level ontologies (TLO) – Part 2: Basic Formal Ontology (BFO), ISO/IEC JTC 1/SC 32, Geneva, CH, 2021. https://www.iso.org/standard/71954.html.

[11] E. Merrell, D. Limbaugh, P. Koch, B. Smith, Capabilities, (2022). https://philpapers.org/rec/MERC-14.

[12] M. Horridge, P.F. Patel-Schneider, OWL 2 Web Ontology Language: Manchester Syntax, World Wide Web Consortium, 2009. http://www.w3.org/TR/owl2-manchester-syntax/.

