Optique

Project N^o:	**FP7-318338**
Project Acronym:	**Optique**
Project Title:	**Scalable End-user Access to Big Data**
Instrument:	**Integrated Project**
Scheme:	**Information & Communication Technologies**

Deliverable D11.5 [extract]
ISO 15926 Part 12: Upper ontology for industrial OBDA applications

Due date of deliverable:	()
Actual submission date:	November 15, 2016

SEVENTH FRAMEWORK PROGRAMME

Start date of the project: **1st November 2012**	Duration: **48 months**
Lead contractor for this deliverable:	UOXF
Dissemination level:	**PU** – Public

Final version

Executive Summary:
ISO 15926 Part 12: Upper ontology for industrial OBDA applications

This document is an extract of deliverable D11.5 of project FP7-318338 (Optique), an Integrated Project supported by the 7th Framework Programme of the European Commission. Full information on this project, including the contents of this deliverable, is available online at http://www.optique-project.eu/.

This document contains the Optique proposal for an OWL 2 DL version of the industry standard ISO 15926, proposed by the Optique project for the ongoing standardisation process of ISO 15926 Part 12.

List of Authors

Johan W. Klüwer (DNV)
Arild Waaler (UiO)
Ernesto Jimenez Ruiz (UOXF, UiO)
Andreas Nakkerud (UiO)

Contents

Chapter 1

Introduction

ISO 15926 is an International Standard for the representation of process industry facility life-cycle information, organised as a series of separately published parts. The most fundamental of these parts of ISO 15926 is its Part 2, which specifies a generic, conceptual data model. Implemented using EXPRESS[1], a standard data modelling language for product data, Part 2 is designed to provide a basis for implementation in a shared database or data warehouse. ISO 15926 Part 2 has had official ISO International Standard status since 2003.

A key assumption is that the data model of Part 2 will be used in conjunction with appropriate reference data. ISO 15926 Part 4 is a so-called ISO Technical Specification (TS) that consists of "Initial reference data". This collection of *core* classes and relations used in relevant industries provides a standard vocabulary with hierarchy: including, for example, "Bolt", "Valve", and "Cable" as physical object types, "Monitoring" and "Testing" as activity types, and "Directive" and "Standard" as document types. The intention is that the classes needed in a specialised project context can be represented as subclasses ("specialisations") of core classes. Reference data is as a rule designed with particular modelling patterns in mind, and a wide range of such patterns is described in the Part 2 documentation, with extensions in subsequent Parts.

The Optique approach is of great interest to users of industrial ontology because it can enable efficient and uniform access to data. The ontology-based approach promises to enable integration across domains, specialist applications, and projects: even as largely the same set of regulations and requirements apply, the project portfolio of any typical enterprise will have inconsistent naming schemes and localised jargon that stands in the way of uniform and efficient access to project data. Optique OBDA promises to make the integrated approach, with standardised reference data and uniform patterns, work also for large and diverse volumes of project data.

With the increasing popularity of OWL, the ISO 15926 communities have gradually started a shift from EXPRESS to OWL. A number of applications of ISO 15926 that implement various parts of ISO 15926 Part 2 and Part 4 are currently in use in industry. In order to support this shift, the community has started a standardisation process with the aim of agreeing on an official OWL version of ISO 15926 Part 2. This OWL rendition of the generic conceptual model is called ISO 15926 Part 12, and is at the time of writing in the process of being proposed as an ISO Draft International Standard (DIS). Figure 1.1 shows the (estimated) completion date of the the different ISO 15926 parts and the establishment of the different modelling languages and technologies relevant to ISO 15926.

Broadly speaking, contributions to Part 12 have come from two communities. One community has concentrated on producing an ontology which deviates as little as possible from the original Part 2; this community is now proposing a version of Part 12 which we in the following will refer to as *Part 12 RDFS* or *the RDFS version*. The RDFS version is in particular designed to support the use of existing reference data with minimal effort, but it makes very limited use of OWL primitives in its proposed modelling patterns. The RDFS version provides a fairly literal recasting and preservation of the patterns that were introduced with the original EXPRESS representations.

[1] `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38047`

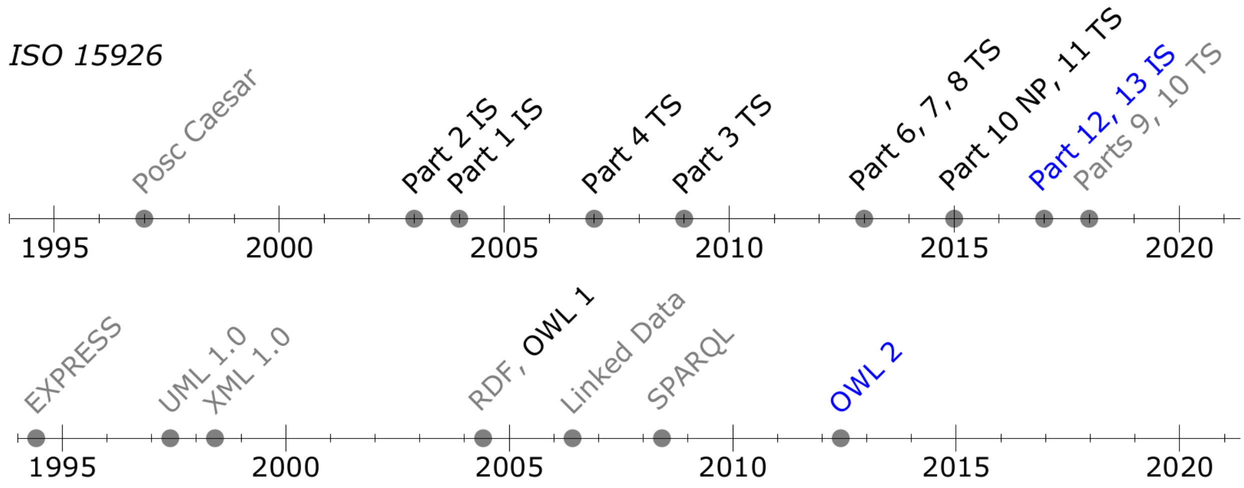


Figure 1.1: Timeline: ISO 15926 and OWL

The Optique project has made an effort to propose a version of Part 12 that supports automated reasoning and use of the Optique platform. This version is in the following referred to as *Part 12 DL* or *the DL version*. The DL version departs from an interpretation of Part 2 ontology that was developed at DNV GL and applied in several industry development projects. This interpretation of ISO 15926 Part 2 is currently in use at the engineering company Aibel, where it supports a successful ontology-based system supporting requirements management and integration of data across large capital projects. Aibel engineers, with support from Optique partner DNVGL, have developed an industrial ontology of a size and scope that we may take to be typical of what industrial users need. The ontology that is currently in production at Aibel has 10-deep subclass hierarchies and tens of thousands of classes, partitioned into hundreds of ontology modules with strictly enforced dependencies. This requires careful modelling, and OWL has been demonstrated to serve excellently as a modelling language for ISO 15926 industrial reference data.

OWL reasoning gives ontology developers the ability to discover implicit facts and hidden inconsistencies. A key learning from the Aibel ontology project is that sound development of an ontology of industrially relevant size and complexity is completely dependent on the support of reasoning services. Assistance from automated reasoning is crucial for managing the complexity of domains and disciplines, and for building a model that can serve a wide range of applications.

Optique has supported the development of ISO 15926 Part 12 in order to secure impact with the industrial users. In short, the industry needs *OWL DL reasoning* to build ontologies (ISO 15926 “reference data”), and *query rewriting* to apply them to project data. The Part 12 DL profile delivers a *practical* basis for ontology-based solutions. Thanks to the availability as an ISO standard, it is readily available for any interested enterprise, and in particular for inter-enterprise solutions. Part 12 DL supports best-practice, reasoning supported modelling. The resulting ontologies will in general be suitable for OBDA applications, for which restricted OWL profiles are required; details of this challenge are discussed in some detail below.

Part 12 DL, as proposed by the Optique project, is summarised in Section 2. Section 3 contains some key modelling patterns of scenarios that are relevant for industrial use of the ontology. Compared to the modelling patterns that come along with the RDFS version, a common denominator of the DL version is that *semantically significant attributes are modelled at the level of individuals through object properties and data properties.* A case in point is given in Figure 3.1, which illustrates a representation of a measurement of the mass of a hammer in kilogram. In addition to the individual representing the hammer, the pattern introduces one individual for the mass, one for the datum, and one for the unit of measure. The three latter individuals are, respectively, members of the classes PhysicalQuantity, InformationObject, and UnitOfMeasure; in the RDFS version the corresponding classes are of type “class of class”, i.e. classes whose members are themselves classes. Also note that in Figure 3.1 the data value representing the mass quantity is a data property of the measurement datum; the RDFS version will capture this attribute through a class membership assertion about the hammer (the hammer is a member of the 4.7 kilogram class).

Following Part 2, the RDFS version encodes semantically significant information in class names rather than in properties of individuals. A case in point is the modelling of a functional physical object. In the RDFS version this is modelled simply as a class that is a subclass of the physical object class. But as long as this is the only definition, there is no way that the RDFS version can be used to infer from attribute values that a given physical object is in fact also functional. In order to support this kind of reasoning, the modelling pattern for the DL version introduces an individual representing the function and another individual representing a suitable activity, as illustrated in Figure 3.2. The final example in Section 3 illustrates how product design can be captured in the DL version, where corresponding representation in the RDFS version will use "class of class" constructs.

The relationship between the OWL 2 QL profile that the Optique platform requires and the OWL 2 language needs to be made clear. While the DL version is designed for use of OWL 2 reasoning tools, it is not obvious that this version will serve the needs of the users of the Optique platform. To this, note that some of the most important restrictions on OWL QL ontologies are also restrictions on OWL DL ontologies. While OWL Full allows both membership and subclass relations between classes, OWL DL allows only the subclass relationship. The same restriction holds on properties. Since OWL QL allows just the same hierarchical structure as OWL DL, the core structure of any OWL DL ontology will also be OWL QL.

There are some major differences between the OWL DL and OWL QL profiles, but most of these have no effect on modelling patterns. A consequence is that any OWL DL ontology can be approximated to an OWL QL ontology. The approximated OWL QL ontology will keep the core structure of the original OWL DL ontology. This in turn will make it easy for users to adapt to the new ontology.

The challenge facing a user of an approximated ontology is that approximated ontologies have reduced inference power. This means that the user may have to include more information in their queries, in order to compensate for statements that were removed from the original ontology. A user of an OWL QL ontology would have to include the same information in a query to get the same results, but would expect this from the beginning.

When users want to use the Optique platform, and they have an OWL DL ontology, then only a slight modification must be made to the ontology for it to be used with Optique. This change will be of a nature where the users can keep their conceptual model of their domain. They can keep the way they think about the core building blocks of ontologies. The only required concession will be writing slightly more complicated queries. The complications come from a need to capture some lost details about connections between classes, properties and data values.

OWL 2 has, however, limitations as to what reasoning support can be provided. Section 4 addresses potential enhancements by means of rules and SPARQL workarounds.

Chapter 2

ISO 15926-12 Upper ontology

In this section we provide an overview of the main entities of the OWL 2 DL version of the ISO 15926 part 12 upper ontology. Figure 2.1 shows the classes, object properties, and data properties included. Appendix A includes the complete mapping between Part 2 and Part 12 and a complete listing of the ontology.

Note that ISO 15926 part 12 is an upper ontology and aims at coordinating ontologies about diverse domains and with different degrees of specificity. Furthermore, it must support collaboration in development of ontologies. Thus ISO 15926 part 12 should contain classes and relationships to cover all relevant domains.

2.1 Classes

We review the classes of primary interest.

PhysicalObject Physical objects are typically the main citizens in an industrial ontology. Objects in this category will typically have functions, be involved in activities, and possess qualities.

Function Some objects have functions that are simple, such as a nut serving to secure a bolt in its place, while others have complex and generic functions, such as a control mechanism or a robot arm. It is common to talk about functions changing over time, which indicate that it is reasonable to represent functions as individuals and to include *Function* as an upper ontology class. *Function* class is inspired by the Basic Formal Ontology (BFO)[1] class "function".

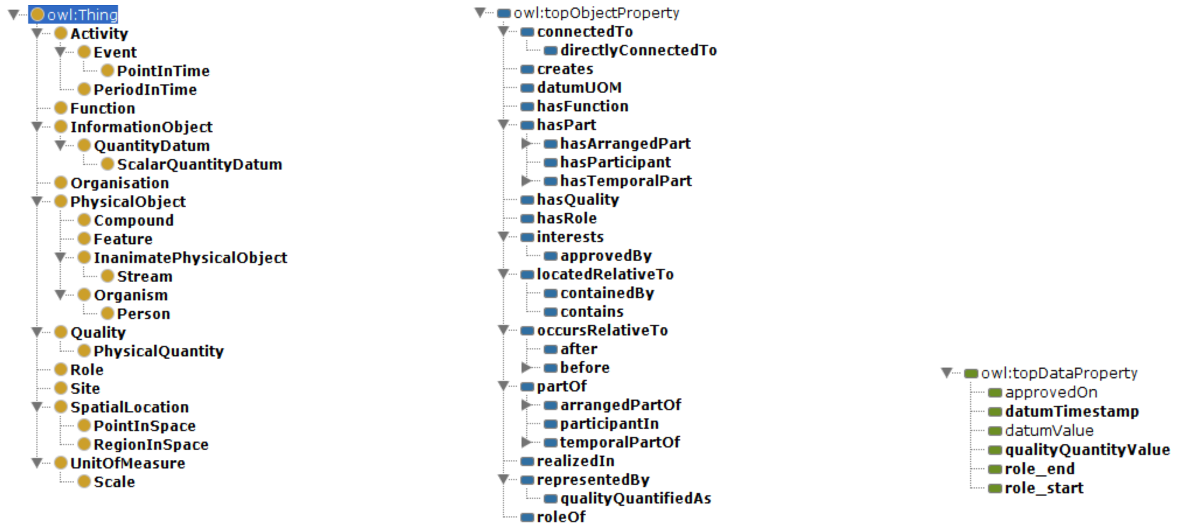
Activity The class *Activity* will group concrete activities or processes like *NailDriving* (i.e. the driving of a nail into a material).

QuantityDatum This class is inspired by the class "measurement datum" of the Information Artefact Ontology (IAO)[2]. The change of wording from "measurement" to "quantity" is intended to support cases where measurement is not involved, such as with nominal values. *QuantityDatumMass*, *QuantityDatumPresssure* and *QuantityDatumTemperature* are examples of possible subclasses of *QuantityDatum*.

ScalarQuantityDatum 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.

Scale This class has units of measure as members, such as `kilogram`, `pascal`, `bar`, `kelvin`, `celsius`.

[1] `http://ifomis.uni-saarland.de/bfo/`
[2] `http://bioportal.bioontology.org/ontologies/IAO`



(a) Classes (b) Object properties (c) Data properties

Figure 2.1: ISO 15926 Part 12 upper ontology: DL version

PhysicalQuantity This class and its superclass *Quality* are directly inspired by corresponding classes included in the DOLCE[3] and BFO upper ontologies. *Mass*, *Pressure* and *Temperature* are examples of possible subclasses of *PhysicalQuantity*.

Role This class is motivated in the Part 2 *role* entity type, and in the same-named BFO class. Part 2 is not very specific about the meaning of roles, but the examples are clear enough. There is still much disagreement in the ontology field about how roles should be understood and modelled.

2.2 Object properties

Object properties play a key role in the ontology since they enable the direct connection between individuals (i.e. class members).

hasFunction This object property will typically connect members of the class *PhysicalObject* (domain) with members of the class *Function* (range).

realizedIn This object property will typically connect members of the class *Function* with members of the class *Activity*.

hasQuality This object property will enable the connection with members of the class *Quality*. Potential subproperties like *hasMass* can define more concrete connections among objects (i.e., with members of the class *Mass*).

qualityQuantifiedAs This relation is inspired by the relation "is quality measured as" of the IAO ontology. The term "quantified" replaces "measured" to support cases where measurement is not involved, as in e.g. estimates. This property allows the relationships of members of the class *Quality* with members of *QuantityDatum*. Additionally one could define the subproperty *qualityMeasuredAs* for the cases where a measurement is involved.

[3] `http://www.loa.istc.cnr.it/old/DOLCE.html`

partOf This property and its inverse *hasPart* define a relationship part-whole and indicates that the part (possible and individual) is a part of the whole (possible and individual). A simple composition is indicated, unless a subtype is instantiated too. This property is typically transitive.

participantIn This property is a subproperty of *partOf* and expresses participation in an *Activity*. Depending on the *participant* one could define additional subproperties like *toolIn* and *agentIn*.

datumUOM Relation to assign unit of measure (class *UnitOfMeasure*) to quantity data (class *QuantityDatum*).

2.3 Data properties

Data properties, unlike object properties, connects individuals to data values of a certain datatype. *Datatypes* are special entities that refer to sets of data values. One could see datatypes as special type of classes, the difference is that the former contain data values such as strings and numbers, rather than individuals. For example, the datatype *integer* could be seen as the class of *all* integer values. Alternative representations of datatypes using classes and individuals face the problem of incompleteness due to the inability of representing all possible allowed values for a given datatypes.

datumValue This relation is inspired by the relation "has measurement value" of the IAO ontology, although in our setting we do not require the value to be necessarily measured (e.g. estimated or nominal values).

qualityQuantityValue This is a super-property for "template" relations that combine a quality and a unit of measure into a simple data property. For instance, "mass in kilograms" can be introduced as such a data property, for expressing the mass of an entity on the kilogram scale.

datumTimestamp Relation for recording the time a (measured) value is taken.

role start/end This relations defines the starting and ending date for which a role or qualification has validity.

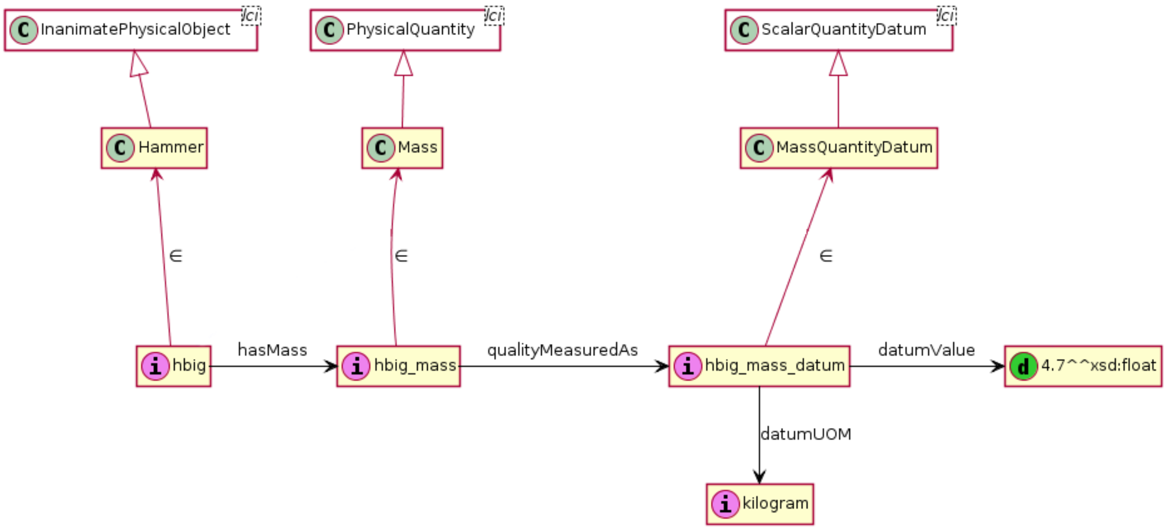


Figure 3.1: Hammer mass representation.

Chapter 3

Modelling patterns for ISO 15926-12

In this section we show relevant examples of modelling patterns in OWL 2 DL for ISO 15926. The modelling patterns aim at, one the one hand, providing means to represent the domain in a comprehensive manner; on the other hand, enabling effective reasoning in practice.

For readability purposes we have removed namespaces. OWL 2 DL axioms are expressed in the Manchester OWL Syntax [6]. Note that the examples are simple and informal for illustration purposes. An *examples* ontology has been implemented to test the correctness of the provided examples.

3.1 Example 1: Physical qualities

In the following we illustrate the main points to model the hammer mass. Figure 3.1 shows the main modelling choices to capture physical quantities. Intuitively, a hammer has a mass and it may have different data associated to its mass, e.g. measured in different points of time or with different units (kilograms, stones). A Hammer and related classes could be formally represented as follows:[1]

```
Class: Hammer
  SubClassOf: hasMass some Mass
Class: Mass
  SubClassOf: qualityQuantifiedAs some MassQuantityDatum
```

[1]Note that, in a real example, one would expect to inherit the restriction with the *hasMass* property from a Superclass instead of being defined for *Hammer*, and similarly for *MassQuantityDatum*.

```
Class: MassQuantityDatum
  SubClassOf:
    datumValue some float and datumUOM some Scale and datumTime some date
```

3.2 Example 2: Necessary and sufficient conditions

In this example we show how to define hammers as small or big. Assume a big hammer is a hammer that weights more than 1 kilogram, while small hammers must weight 1 kilogram or less. This could be formally represented as follows:

```
Class: BigHammer
  SubClassOf: hasMass some (qualityQuantifiedAs
    some (datumUOM value kilogram and datumValue some float[> 1]))

Class: SmallHammer
  SubClassOf: hasMass some (qualityQuantifiedAs
    some (datumUOM value kilogram and datumValue some float[<= 1]))
```

The above two OWL 2 axioms represent necessary but not sufficient conditions for establishing class membership of an individual. For example, the individuals *hbig* and *hsmall* with measured weights of 4.7 and 0.3 kg, respectively, as declared below, will not be classified as *BigHammer*, respectively *SmallHammer*, as one would expect.

```
Individual: hbig
  Types: Hammer
  Facts: hasMass hbig_mass

Individual: hbig_mass
  Types: Mass
  Facts: qualityMeasuredAs hbig_mass_datum

Individual: hbig_mass_datum
  Types: MassMeasurementDatum
  Facts: datumUOM kilogram, datumValue 4.7f

Individual: hsmall
  Types: Hammer
  Facts: hasMass hsmall_mass

Individual: hsmall_mass
  Types: Mass
  Facts: qualityMeasuredAs hsmall_mass_datum

Individual: hsmall_mass_datum
  Types: MassMeasurementDatum
  Facts: datumUOM kilogram, datumValue .3f
```

In order to enable the desired inference, sufficient conditions are also required. This could easily be achieved by declaring *BigHammer* as *EquivalentTo* the restriction instead of *SubClassOf*. An alternative would be to add a reversed *SubClassOf* axioms (i.e. the other side of the equivalence).

```
Class: hasMass some (qualityQuantifiedAs
      some (datumUOM value kilogram and datumValue some float[> 1]))
  SubClassOf: BigHammer

Class: hasMass some (qualityQuantifiedAs
      some (datumUOM value kilogram and datumValue some float[<= 1]))
  SubClassOf: SmallHammer
```

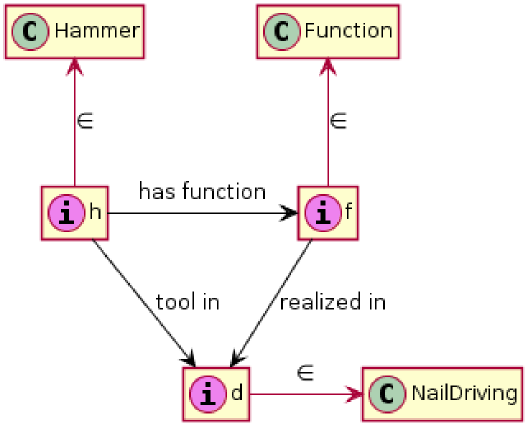


Figure 3.2: Hammer function example.

Apart from enabling additional inferences, sufficient conditions have typically a representation closer to *rules* which will enhance OWL 2 DL reasoning (see Section 4). Necessary conditions may also be considered as restrictions over the data (i.e. integrity constraints) which could also be represented as rules (see Section 4).

3.3 Example 3: Functions

Most of the physical things that we wish to describe in a store of industrial data will have a *function* – an intended purpose. This includes structural elements of a factory, equipment, and instruments.

A description of function could look as follows: "A Hammer's function is realised precisely when it is used as a tool to drive a nail". The shape of the sentence can guide us to a modelling pattern for an ontology-based representation: "A Hammer x has a function that is realised in nail-driving activities where x has the tool role". Figure 3.2 shows the basic pattern. A hammer h has a function f which is realised in the nail-driving activity d. Ensuring that hammer functions are only realised in nail driving processes where the hammer is active as a tool is clearly important (i.e. the link between h and d). This cannot be ensured within OWL 2 but the use of rules can help in this regard (see Section 4).

3.4 Example 4: Conformance with requirements

In this example, we present the generic requirements of an *Electric Motor* at design level to a detailed specification to be installed. The Component specification (*Electric Motor ABCD*) requires at least 850 watts of output power. The *ACME A model* delivers 900 watts and is suitable, but *ACME B* delivers only 800 watts, as an example of a non-conformant choice. We assume a is the individual that represents our motor during design, and that $a1$, $a2$, and $a3$ are replaceable individuals installed to fill the role of a in the assembly. $a1$ and $a2$ are concrete types of the *ACME A model* while $a2$ is of type *ACME B*. Figure 3.3 summarises the defined classes and individuals. The component requirement and product classes would be defined as follows:

```
Class: ElMotorABCD
  SubClassOf: ElMotor and
    power_watts only float[>= 850] and power_watts some float[>= 850]
Class: ElMotorACME_A
  SubClassOf: ElMotor and power_watts value 900f
Class: ElMotorACME_B
  SubClassOf: ElMotor and power_watts value 800f
```

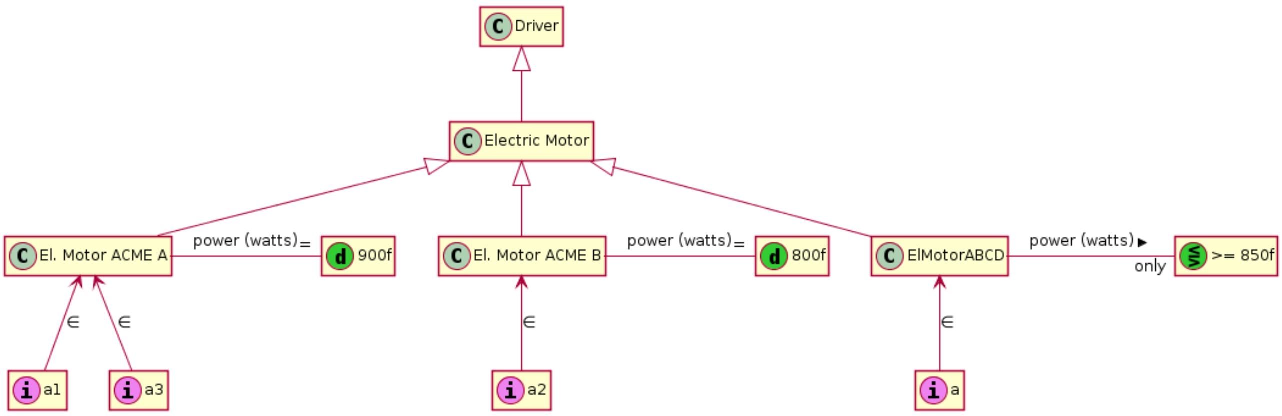


Figure 3.3: Requirements Electric Motor.

Using automated reasoning we can check whether the requirements laid down in a design are satisfied by the installed parts. In complex cases, we benefit from the reasoner's ability to find not only obvious clashes, but also any implicit conflicts that may be very difficult to identify without the help of automated reasoning. There are different solutions to check conformance requirements:

- *Checking emptiness or disjointness* between the component specification class and the classes describing the concrete specifications of a model. For example the intersection between *ElMotorABCD* and *ElMotorACME_A* is non empty since the *ACME A model* satisfies the requirements (delivers 900 watts) while the intersection between *ElMotorABCD* and *ElMotorACME_B* is empty since *ACME A model* does not meet the requirements (delivers only 800 watts).

- *Individual substitution.* This can be done by selecting "concrete" individuals of a model and substituting them by the targeted design objects. For the example given, we substitute the replaceable parts $a1$, $a2$ and $a3$ for the design object a. The effect of substitution is that we combine all the requirements of the design with all the characteristics of the product specimens. Substitution can be simulated by adding statements of the type:

```
Individual: a
  SameAs: a1
```

  If there is a conflict, the reasoner will discover an inconsistency. The difference with respect to the previous solution is that the concrete individuals of a model may bring additional characteristics to meet the design requirements.

- *Checking membership.* Alternatively, instead of finding conflicts between the requirements and the concrete products, one could try to classify the concrete product individuals and model specifications under the component requirement specification. To this end, as in Example 2, sufficient conditions are required. Adding the following sufficient condition to our example would classify *ElMotorACME_A* under *ElMotorABCD* and thus the replaceable parts $a1$ and $a3$ will also be members of *ElMotorABCD*.

```
Class: ElMotor and power_watts some float[>= 850]
  SubClassOf: ElMotorABCD
```

Chapter 4

Enhancing OWL 2 DL reasoning

Reasoning with OWL 2 DL ontologies can be enhanced using (datalog) rules (e.g. SWRL[1]) and SPARQL workarounds. State-of-the-art reasoners like HermiT [4] and RDFox [10] provide support for SWRL rules in combination with OWL 2 ontologies or OWL 2 RL ontologies for the case of RDFox. -Ontop-, the query rewriting system used in Optique, also supports the use of rules [12]. In this sections we present some examples of potential enhancements.

Shortcuts The model of physical qualities exemplified in Section 3.1 is very complete and detailed since, for example, the mass can be measured at different points of time and using different unit of measure. However, it may be practical to infer a direct relationship (i.e. a shortcut) between the object and the mass value with the corresponding unit of measure (see Figure 4.1). This can be achieved with the following rule:

```
hasMass(?x,?y), qualityMeasuredAs(?y,?z), datumUOM(?z,kilogram), datumValue(?z,?u) ->
           hasMass_in_kilogram(?x,?u)
```

Note that part of the (semantic) information of the model is now represented in the novel data property name *hasMass_ in_ kilogram*.

Checking completeness Conformance with a design requires not violating one of the requirements (e.g. delivering less power than required) but also being complete. Identifying incomplete products (e.g. those without delivered power specification) as not suitable is naturally captured by integrity constraints. Integrity constraints, however, are not supported in OWL 2 since it requires (non-monotonic) reasoning with negation-as-failure. A product not delivering power, using the OWL 2 semantics, does not necessarily violate the requirement specification. OWL 2 reasoning assumes that, although the data is not in the knowledge base, it may exist somewhere else (e.g. the data is unknown or unspecified).

There have been several proposals to extend OWL 2 with integrity constraints [9, 11, 7]. In these approaches, the ontology developer explicitly designates a subset of the OWL 2 axioms as constraints. Similarly to constraints in databases, these axioms are used as checks over the given data and do not participate in query answering once the data has been validated. The specifics of how this is accomplished semantically differ amongst each of the proposals; however, all approaches largely coincide if the standard axioms are in OWL 2 RL.

For example, Table 4.1 provides the set of OWL 2 axioms considered as constraints together with their translation into *rules with stratified negation* suggested in [7].[2] The translation assigns a unique id to each individual axiom marked as an integrity constraint in the ontology, and it introduces predicates not occurring in the ontology in the heads of all rules. Constraint violations are recorded using the fresh predicate *Violation* relating individuals to constraint axiom ids.

In our example to identify *Electric Motors* with unspecified or unknown delivered power as incomplete, we could consider the following OWL 2 axiom as an integrity constraint:

[1] `https://www.w3.org/Submission/SWRL/`

[2] This selection and translation of OWL 2 axioms is also used in Section **??**

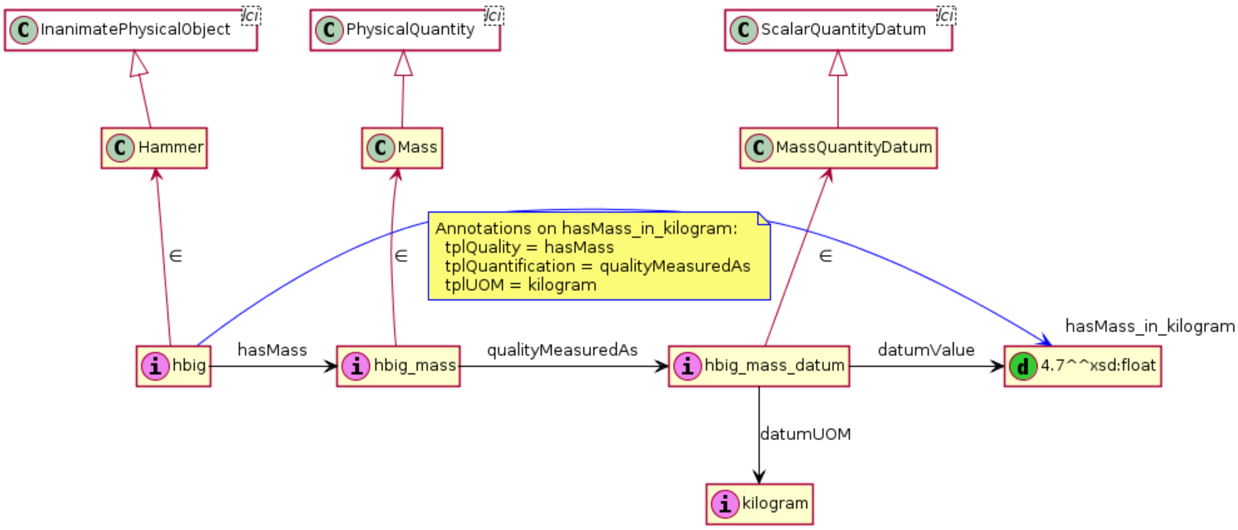


Figure 4.1: Shortcut for the hammer mass representation.

OWL Axiom	Datalog rules
A SubClassOf: R some B	$R\_B(?x) \leftarrow R(?x, ?y) \ \wedge \ B(?y)$ and $Violation(?x, \alpha) \leftarrow A(?x) \ \wedge \ \textbf{not} \ R\_B(?x)$
A SubClassOf: R value b	$Violation(?x, \alpha) \leftarrow A(?x) \ \wedge \ \textbf{not} \ R(?x, b)$
R Characteristics: Functional	$R\_2(?x) \leftarrow R(?x, ?y_1) \ \wedge \ R(?x, ?y_2) \ \wedge \ \textbf{not} \ owl{:}sameAs(?y_1, ?y_2)$ and $Violation(?x, \alpha) \leftarrow R\_2(?x)$
A SubClassOf: R max n B	$R\_(n{+}1)\_B(?x) \leftarrow \bigwedge_{1 \le i \le n+1} (R(?x, ?y_i) \wedge B(?y_i)) \bigwedge_{1 \le i < j \le n+1} (\textbf{not} \ owl{:}sameAs(?y_i, ?y_j))$ and $Violation(?x, \alpha) \leftarrow A(?x) \ \wedge \ R\_(n{+}1)\_B(?x)$
A SubClassOf: R min n B	$R\_n\_B(?x) \leftarrow \bigwedge_{1 \le i \le n} (R(?x, ?y_i) \wedge B(?y_i)) \bigwedge_{1 \le i < j \le n} (\textbf{not} \ owl{:}sameAs(?y_i, ?y_j))$ and $Violation(?x, \alpha) \leftarrow A(?x) \ \wedge \ \textbf{not} \ R\_n\_B(?x)$

Table 4.1: Constraints axioms as rules. All entities are named, $n \geq 1$, and α is the unique id for the given constraint. Note that the axioms involving the property R apply both for the Object property and Data property cases of R.

```
Class: ElMotor
  SubClassOf: power_watts some float
```

Analogously to the translation suggested in Table 4.1, the above axiom could be translated into the following rule with negation:

```
ElMotor(?x), not power_watts(?x,?y) -> Incomplete_ElMotor(?x)
```

The (fresh) predicate *Incomplete_ ElMotor* could also be defined as a disjoint class with the component requirement specification class to enhance reasoning.

Reasoning with integrity constraints (e.g., negation-as-failure) can be currently implemented with any RDF triple store with support for OWL 2 RL and stratified negation (e.g., RDFox [10]), as well as on top of generic rule inference systems (e.g., IRIS [1]).

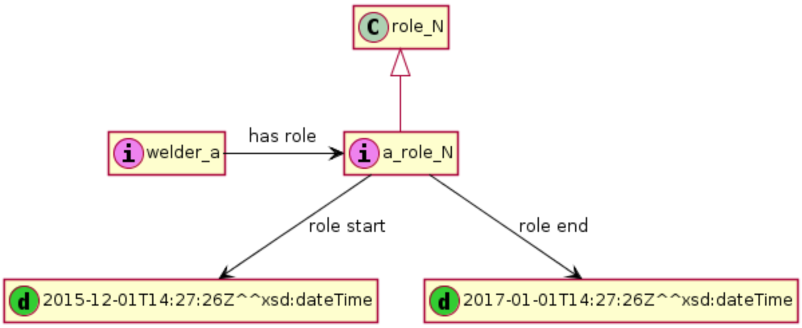


Figure 4.2: Starting and ending times for a role.

Ensuring correctness of functions Functions of a *PhysicalObject* should only be realised in activities where the object is an active participant. For example, ensuring that hammer functions (see Section 3.3) are only realised in nail driving processes where the hammer is active as a tool can easily be defined with the following rule:

```
Hammer(?x), hasFunction(?x,?y), realizedIn(?y,?z) -> toolIn(?x,?z)
```

Detecting critical and optimal conditions Detecting if, for example, a stream has a critical or optimal pressure and/or temperature can be done similarly to detecting if a hammer is small or big (see examples in Section 3.2). For example, one could define that a temperature above 850 Celsius, a pressure higher than 15 bar, or any combination of temperature above 700 Celsius and pressure above 13 bar, as critical. The latter however requires to take into account time stamps to consider only combinations of temperature measurements and pressure measurements that are close enough in time.

We advocate for not including temporal description logic within the ontology. Reasoning about time dependent concepts rapidly becomes an undecidable problem for small temporal extension of OWL 2 profiles like EL (e.g. [5].) Querying about time can be done at the application level using rules and SPARQL queries where we can define the meaning of two time stamps being close in time using the BIND and FILTER constructs.

Validity of roles Detecting if, for example, a welder is qualified to perform a task that requires a *valid role* for a given date or interval of dates can easily be achieved using rules. One could also encode this information within the ontology but it is rather application/query dependant so we advocate the use of dedicated rules or SPARQL queries to check for role validity. Figure 4.2 encodes the data associated to a role for a concrete agent (i.e. *welder_a*).

Checking if an *Agent* is qualified to perform an *Activity* happening at a given date according to the validity of his/her role can be encoded in the following general rule:

```
Activity(?x), has_date(?x,?t), Agent(?y), has_role(?y,?z), Role(?z),
 role_start(?z,?t1), role_end(?z,?t2), ?t1 < ?t, ?t < ?t2 ->
           qualified-to-perform-activity(?y,?x)
```

Note that the use of such rules is more flexible and elegant than encoding in the ontology (infinitely many possible) classes such as *Qualified_welder_at_date_X*.

4.1 Reasoning with OWL 2 profiles

Arbitrary rules and OWL 2 ontologies may lead to undecidability. Furthermore, reasoning with data and OWL 2 DL ontologies is expensive and may lead to tractability problems as the data grows. The use of

OWL 2 profiles plays a key role in this regard.[3]

As mentioned in Section 1 there are means to translate OWL 2 DL ontologies to a specific profile since the core components of an ontology are shared by all the profiles (e.g., class and property hierarchies, domain and ranges). Furthermore, information that cannot be captured in the profiles can still be captured in dedicated rules, SPARQL queries or even in the mappings connecting ontology terms to relational database data.

The OWL 2 RL profile enables efficient reasoning with large amounts of data since OWL 2 RL ontologies have a direct translation to datalog rules. Furthermore the use of OWL 2 RL ontologies also enables the reasoning with integrity constraints as described above. The complexity of answering conjunctive queries in the RL profile is PTIME-complete with respect to the size/complexity of the ontology and the data.[4] The complexity of answering conjunctive queries in the OWL 2 QL profile is NLOGSPACE-complete and LOGSPACE with respect to the size/complexity of the ontology and the data, respectively.[5]

In an OBDA scenario like in Optique, where the data lives in a relational database (e.g., it cannot be materialised as ontology triples due to security, size, or other constraints of the operational scenario), OWL 2 QL ontologies are required to allow the rewriting of ontology queries (e.g., SPARQL) to the target database queries (e.g., SQL).[6] [7] However there is an exponential blow-up in the rewriting that is unavoidable in OWL 2 QL. This blow-up is due to the combination of rich ontology hierarchies, existentials, and the complexity of mappings between the ontology and the relational database [8], which may lead to a very large number of SQL queries.

Modelling patterns should take into account the compromise between expressive power and tractability, and find solutions that are both effective and efficient in practice. For example, large and deep ontology hierarchies may not be suitable in an OBDA scenario as described above. Furthermore, large hierarchies in an OBDA scenario will typically imply larger set of mappings that need to be created and maintained.

State-of-the-art triple stores allow for efficient storage and retrieval of relatively large amounts of data.[8] While in-memory triple stores are limited by the available RAM, RDFox, as one example, is economical with memory and can store up to 1.5 billion triples in 50 GB of RAM. If data is very large the only solution may be to use a traditional relational database and commit to an OBDA approach where an OWL 2 QL ontology is required.

[3] See OWL 2 computational properties here: `https://www.w3.org/TR/owl2-profiles/#Computational_Properties`

[4] PTIME is the class of problems solvable by a deterministic algorithm using time that is at most polynomial in the size of the input. PTIME is often referred to as tractable, whereas the problems in the classes above are often referred to as intractable.

[5] LOGSPACE is the class of problems solvable by a deterministic algorithm using space that is at most logarithmic in the size of the input. NLOGSPACE is the nondeterministic version of this class.

[6] The language underlying OWL 2 QL has the first-order (FO) rewritability property [3].

[7] Some recent works are pushing information from the ontologies to the mappings to allow ontologies beyond OWL 2 QL [2].

[8] `https://www.w3.org/wiki/LargeTripleStores`

Bibliography

[1] Barry Bishop and Florian Ficsher. Iris - integrated rule inference system. In *Workshop on Advancing Reasoning on the Web*, 2008.

[2] Elena Botoeva, Diego Calvanese, Valerio Santarelli, Domenico Fabio Savo, Alessandro Solimando, and Guohui Xiao. Beyond OWL 2 QL in OBDA: rewritings and approximations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 921–928, 2016.

[3] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *JAR*, 39(3):385–429, 2007.

[4] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. Hermit: An OWL 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014.

[5] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Roman Kontchakov. On decidability and tractability of querying in temporal EL. In *Proceedings of the 29th International Workshop on Description Logics, Cape Town, South Africa, April 22-25, 2016.*, 2016.

[6] Matthew Horridge, Nick Drummond, John Goodwin, Alan L. Rector, Robert Stevens, and Hai Wang. The Manchester OWL Syntax. In *OWLED*, 2006.

[7] Evgeny Kharlamov, Bernardo Cuenca Grau, Ernesto Jiménez-Ruiz, Steffen Lamparter, Gulnar Mehdi, Martin Ringsquandl, Yavor Nenov, Stephan Grimm, Mikhail Roshchin, and Ian Horrocks. Capturing industrial information models with ontologies and constraints. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, pages 325–343, 2016.

[8] Davide Lanti, Martín Rezk, Guohui Xiao, and Diego Calvanese. The NPD benchmark: Reality check for OBDA systems. In *Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015, Brussels, Belgium, March 23-27, 2015.*, pages 617–628, 2015.

[9] Boris Motik, Ian Horrocks, and Ulrike Sattler. Bridging the gap between OWL and relational databases. *J. Web Sem.*, 7(2):74–89, 2009.

[10] Boris Motik, Yavor Nenov, Robert Piro, Ian Horrocks, and Dan Olteanu. Parallel Materialisation of Datalog Programs in Centralised, Main-Memory RDF Systems. In *AAAI*, pages 129–137, 2014.

[11] Jiao Tao, Evren Sirin, Jie Bao, and Deborah L. McGuinness. Integrity constraints in OWL. In *AAAI*, 2010.

[12] Guohui Xiao, Martin Rezk, Mariano Rodriguez-Muro, and Diego Calvanese. Rules and ontology based data access. In *Web Reasoning and Rule Systems - 8th International Conference, RR 2014, Athens, Greece, September 15-17, 2014. Proceedings*, pages 157–172, 2014.

Glossary

BFO	Basic Formal Ontology
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
IAO	Information Artefact Ontology
IEC	International Electrotechnical Commission
ISA	International Society of Automation
ISO	International Organization for Standardization
NPD	Norwegian Petroleum Directorate
O&M	Ontology and Mapping
OBDA	Ontology-based Data Access
OWA	Open-world Assumption
OWL	Web Ontology Language
SPARQL	SPARQL Protocol and RDF Query Language
STARQL	Streaming and Temporal ontology Access with a Reasoning-based Query Language
RDF	Resource Description Framework
RODI	Relational-to-Ontology Data Integration Scenarios
W3C	World Wide Web Consortium

Appendix A

ISO 15926 Part 12 ontology: DL profile

This appendix reports the current version of the document describing the DL profile ontology version of ISO 15926 part 12.

A.1 Map of entity types from the Part 12 DL profile to Part 2

The following table corresponds to that included with the Community Draft (CD) version of Part 12. We see using OWL modelling patterns and SKOS for meta-classes allows us to considerably reduce the number of entity types.

For some entries in the table we recommend to «implement as reference data». Individual cases may be found to be generic enough that they ought to be included in Part 12 itself; examples are «molecule» and «responsibility».

In the current preliminary proposal, 168 of the 202 Part 2 entity types are found not to be needed, marked with «–». They fall into one of seven categories:

- 77 are «class of class» entity types that can be handled using the SKOS vocabulary,
- 7 are modal notions and «not suitable» for a description logic ontology,
- 12 are determined to be *out of scope*, in line with the CD version,
- 11 are replaced by application of XSD data types built into OWL,
- 19 are recommended to not be part of Part 12 itself, but moved to a reference data library,
- 12 are found to be «not needed», typically entity types that are OWL native or that were only included in Part 2 due to EXPRESS specific constraints,
- 32 are marked with «use OWL» – the entity types can be represented using OWL modelling patterns.

For 34 (= 202 - 168) entity types, we have a fairly direct match in resources proposed for the Part 12 DL ontology. Note that that ontology has more than 34 resources defined. This is generally (1) to introduce new resources required to support the DL style of modelling, or (2) because Part 2 only has the «class of N» variant of an entity type explicitly defined, where the corresponding «N» type is given in the DL ontology.

ISO 15926-2 entity	LIS-12-DL	DL note
activity	Activity	
actual_individual	–	modal, not suitable for DL
arranged_individual	–	use OWL: hasArrangedPart
arrangement_of_individual	–	use OWL: hasArrangedPart (twice)
assembly_of_individual	–	use OWL: hasAssembledPart

Continued on next page

Continued from previous page

ISO 15926-2 entity	LIS-12-DL	DL note
beginning	begins, hasBeginning	
cause_of_event	–	use OWL: causes
class_of_atom	–	use SKOS; implement as reference data
class_of_biological_matter	–	use SKOS; implement as reference data
class_of_cause_of_beginning_of_class_of_individual	–	use SKOS; see begins
class_of_cause_of_ending_of_class_of_individual	–	use SKOS; see ends
class_of_class_of_individual	–	use SKOS
class_of_composite_material	–	use SKOS; implement as reference data
class_of_compound	–	use SKOS; see Compound
class_of_feature	–	use SKOS; see Feature
class_of_functional_object	–	use SKOS; see examples:Function
class_of_inanimate_physical_object	–	use SKOS; see InanimatePhysicalObject
class_of_individual	–	use SKOS
class_of_molecule	–	use SKOS; implement as reference data
class_of_organism	–	use SKOS; see Organism
class_of_organisation	–	use SKOS; see Organisation
class_of_particulate_material	–	use SKOS; implement as reference data
class_of_person	–	use SKOS; see Person
class_of_sub_atomic_particle	–	use SKOS; implement as reference data
composition_of_individual	isPartOf, hasPart	
connection_of_individual	connectedTo	
containment_of_individual	contains, containedBy	
crystalline_structure	–	implement as reference data
direct_connection	directlyConnectedTo	
ending	ends, hasEnd	
event	Event	
feature_whole_part	featureOf, hasFeature	
functional_physical_object	–	use OWL: hasFunction
indirect_connection	–	use OWL: directlyConnectedTo, negation
materialized_physical_object	–	modal, not suitable for DL
participation	participantIn, hasParticipant	
period_in_time	PeriodInTime	
phase	Phase	
physical_object	PhysicalObject	
point_in_time	PointInTime	
possible_individual	owl:Thing	
relative_location	locatedRelativeTo	
spatial_location	SpatialLocation	
status	–	use SKOS; implement as reference data
stream	Stream	TODO implement as reference data?
temporal_bounding	hasTemporalBound	
temporal_sequence	occursRelativeTo	
temporal_whole_part	hasTemporalPart	restricted to Activity domain/range
whole_life_individual	–	modal, not suitable for DL
class_of_class_of_definition	–	use SKOS (on annotation properties)
class_of_class_of_description	–	use SKOS (on annotation properties)
class_of_definition	–	use SKOS (on annotation properties)
class_of_description	–	use SKOS (on annotation properties)
definition	skos:definition	
description	skos:scopeNote	
involvement_by_reference	–	could add «refers to» annotation property
class_of_class_of_information_representation	–	use SKOS; see InformationObject
class_of_class_of_representation	–	use SKOS; see InformationObject
class_of_class_of_representation_translation	–	use SKOS; implement in reference data
class_of_class_of_responsibility_for_representation	–	use SKOS; see InformationObject
class_of_class_of_usage_of_representation	–	use SKOS; see InformationObject
class_of_information_object	–	use SKOS; see InformationObject
class_of_information_presentation	–	use SKOS; see InformationObject
class_of_information_representation	–	use SKOS; see InformationObject
class_of_representation_of_thing	–	use SKOS; see InformationObject
class_of_representation_translation	–	use SKOS; see InformationObject
class_of_responsibility_for_representation	–	use SKOS; see InformationObject
class_of_usage_of_representation	–	use SKOS; see InformationObject
document_definition	–	reference data InformationObject subclass
EXPRESS_string	–	use XSD data type
language	–	use RDF language tag or reference data
representation_form	–	implement in reference data (file format)
representation_of_thing	representedBy	

Continued on next page

Continued from previous page

ISO 15926-2 entity	LIS-12-DL	DL note
responsibility_for_representation	–	implement in reference data (roles)
usage_of_representation	–	implement in reference data (roles)
boundary_of_number_space	–	out of scope (as in CD)
class_of_dimension_for_shape	–	out of scope (as in CD)
class_of_shape	–	out of scope (as in CD)
class_of_shape_dimension	–	out of scope (as in CD)
coordinate_system	–	out of scope (as in CD)
dimension_of_individual	–	out of scope (as in CD)
dimension_of_shape	–	out of scope (as in CD)
individual_dimension	–	out of scope (as in CD)
property_for_shape_dimension	–	out of scope (as in CD)
property_space_for_class_of_shape_dimension	–	out of scope (as in CD)
shape	–	out of scope (as in CD)
shape_dimension	–	out of scope (as in CD)
class_of_class_of_identification	–	use SKOS
class_of_identification	–	use SKOS
class_of_left_namespace	–	not needed
class_of_namespace	–	not needed
class_of_right_namespace	–	not needed
identification	skos:prefLabel	use SKOS (as in CD)
left_namespace	–	not needed
namespace	–	not needed
right_namespace	–	not needed
class_of_intended_role_and_domain	–	modal, not suitable for DL
class_of_possible_role_and_domain	–	modal, not suitable for DL
intended_role_and_domain	–	modal, not suitable for DL
possible_role_and_domain	–	modal, not suitable for DL
arithmetic_number	–	use XSD data type
class_of_functional_mapping	–	use SKOS (on object/data properties)
class_of_isomorphic_functional_mapping	–	use SKOS (on object/data properties)
class_of_number	–	use SKOS, or OWL value ranges
enumerated_number_set	–	use OWL nominals
functional_mapping	–	use owl:FunctionalProperty
integer_number	–	use XSD data type
lower_bound_of_number_range	–	use OWL data range
multidimensional_number	–	implement in reference data
multidimensional_number_space	–	implement in reference data
number_range	–	use OWL data range
number_space	–	use OWL data range
real_number	–	use XSD data type
upper_bound_of_number_range	–	use OWL data range
abstract_object	–	not needed/implement as reference data
cardinality	–	use OWL cardinality constraints
class	–	not needed
class_of_class_of_relationship	–	use SKOS (on object/data properties)
class_of_class_of_relationship_with_signature	–	use SKOS (on object/data properties)
class_of_classification	–	not needed, OWL has rdf:type only
class_of_relationship	–	use OWL object/data properties
class_of_relationship_with_related_end_1	–	use OWL object/data properties
class_of_relationship_with_related_end_2	–	use OWL object/data properties
class_of_relationship_with_signature	–	use OWL object/data properties
class_of_specialization	–	not needed, OWL has rdfs:subClassOf only
classification	–	use OWL (rdf:type)
difference_of_set_of_class	–	use OWL union and negation
enumerated_set_of_class	–	use OWL nominals
intersection_of_set_of_class	–	use OWL intersection
multidimensional_object	–	use OWL list ontology – if required
other_relationship	–	use OWL object/data properties
participating_role_and_domain	–	not needed
relationship	–	not needed
role	Role	(this is left out of CD version)
role_and_domain	Role	use OWL «class from role» pattern
specialization	–	use rdfs:subClassOf
specialization_by_domain	–	use rdfs:subClassOf
specialization_by_role	–	use rdfs:subClassOf
thing	owl:Thing	
union_of_set_of_class	–	use OWL union
class_of_individual_used_in_connection	–	use OWL constraints as usual

Continued on next page

Continued from previous page

ISO 15926-2 entity	LIS-12-DL	DL note
individual_used_in_connection	–	use OWL constraints as usual
class_of_abstract_object	–	use SKOS
class_of_activity	–	use SKOS
class_of_arranged_individual	–	use SKOS
class_of_arrangement_of_individual	–	use SKOS
class_of_assembly_of_individual	–	use SKOS
class_of_cause_of_beginning_of_class_of_individual	–	use SKOS
class_of_cause_of_ending_of_class_of_individual	–	use SKOS
class_of_class	–	use SKOS
class_of_class_of_composition	–	use SKOS
class_of_composition_of_individual	–	use SKOS
class_of_connection_of_individual	–	use SKOS
class_of_containment_of_individual	–	use SKOS
class_of_direct_connection	–	use SKOS
class_of_event	–	use SKOS
class_of_feature_whole_part	–	use SKOS
class_of_indirect_connection	–	use SKOS
class_of_indirect_property	–	use SKOS
class_of_involvement_by_reference	–	use SKOS
class_of_multidimensional_object	–	use SKOS
class_of_participation	–	use SKOS
class_of_period_in_time	–	use SKOS
class_of_point_in_time	–	use SKOS
class_of_property	–	use SKOS
class_of_property_space	–	use SKOS
class_of_relative_location	–	use SKOS
class_of_scale	–	use SKOS
class_of_status	–	use SKOS
class_of_temporal_sequence	–	use SKOS
class_of_temporal_whole_part	–	use SKOS
approval	approvedBy, approvedOn	
class_of_approval	–	use SKOS
class_of_approval_by_status	–	use SKOS
class_of_assertion	–	use OWL annotated axioms
class_of_lifecycle_stage	–	use SKOS
class_of_recognition	–	use SKOS
lifecycle_stage	interests	see explanatory note for why this is OK
recognition	–	use OWL pattern with ScalarQuantityDatum
boundary_of_property_space	–	use OWL data ranges (complex)
class_of_scale_conversion	–	use OWL pattern with UOM reference data
comparison_of_property	–	use OWL pattern/see `https://www.w3.org/TR/owl2-dr-linear/`
enumerated_property_set	–	use OWL nominals with ScalarQuantityDatum
indirect_property	–	use OWL data ranges with ScalarQuantityDatum
lower_bound_of_property_range	–	use OWL data ranges with ScalarQuantityDatum
multidimensional_property	–	use OWL pattern with ScalarQuantityDatum
multidimensional_property_space	–	use OWL and/or approximation with ScalarQuantityDatum
multidimensional_scale	–	use OWL pattern with ScalarQuantityDatum
property	PhysicalQuantity indivdual	DL profile also provides the more generic Quality
property_quantification	qualityQuantifiedAs	use qualityMeasuredAs for measured quantities
property_range	PhysicalQuantity subclass	use OWL data ranges with ScalarQuantityDatum
property_space	PhysicalQuantity subclass	
representation_of_Gregorian_date_and_UTC_time	–	use XSD data type
scale	Scale	
single_property_dimension	PhysicalQuantity subclass	
specialization_of_individual_dimension_from_property	–	use OWL rdfs:subClassOf on PhysicalQuantity

Continued on next page

Continued from previous page

ISO 15926-2 entity	LIS-12-DL	DL note
upper_bound_of_property_range	–	use OWL data ranges with ScalarQuantityDatum
class_of_EXPRESS_information_representation	–	use XSD data types
EXPRESS_binary	–	use XSD data type
EXPRESS_Boolean	–	use XSD data type
EXPRESS_integer	–	use XSD data type
EXPRESS_logical	–	use XSD data type
EXPRESS_real	–	use XSD data type

A.2 Ontology listing

The following is a listing of the ISO 15926 Part 12, DL profile ontology in OWL Manchester Syntax.

```
Prefix: : <http://standards.iso.org/iso/15926/-12/tech/ontology/DL-profile#>
Prefix: dc: <http://purl.org/dc/elements/1.1/>
Prefix: lci: <http://standards.iso.org/iso/15926/>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: pav: <http://purl.org/pav/>
Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix: skos: <http://www.w3.org/2004/02/skos/core#>
Prefix: xml: <http://www.w3.org/XML/1998/namespace>
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>



Ontology: <http://standards.iso.org/iso/15926/-12/tech/ontology/DL-profile>


Annotations:
   rdfs:comment "This ontology contains the DL profile of ISO 15926-12, which represents ISO 15926-2 in
       →OWL 2.",
   rdfs:label "ISO 15926-12, DL profile",
   owl:versionInfo "Date: 2016-11-12"

AnnotationProperty: lci:CD

   Annotations:
      rdfs:label "CD comment",
      rdfs:comment "Annotation property for use in Part 12 DL profile drafts, to comment on how the
          →annotated resource relates to (typically, deviates from) the RDFS profile of the CD version of
          →Part 12."


AnnotationProperty: lci:definitionPart2

   Annotations:
      rdfs:label "definitionPart2",
      rdfs:comment "Annotation property for recording definitions of entity types given in the original
          →Part 2 of ISO 15926, where the Part 2 and Part 12 entities are equivalent in meaning for all
          →practical purposes."


AnnotationProperty: lci:deprecatedPart2

   Annotations:
      rdfs:comment "Annotation property for recording annotations of related entity types given in the
          →original Part 2 of ISO 15926, where Part 2 entity type is considered not suitable for a DL
          →rendering.",
      rdfs:label "deprecated Part 2"
```

```
AnnotationProperty: lci:equivalentPart2

    Annotations:
        rdfs:comment "Annotation property for recording annotations of entity types given in the original
            →Part 2 of ISO 15926, where the Part 12 entity type is equivalent in meaning to that of Part 2.
            →Metaclass (\"class of ...\") entity definitions are included.",
        rdfs:label "equivalent Part 2"


AnnotationProperty: lci:remodelledPart2

    Annotations:
        rdfs:comment "Annotation property for recording annotations of entity types given in the original
            →Part 2 of ISO 15926, where the Part 12 entity type provides the same expressive power using a
            →different category, typically by recasting a Part 2 class as a constraint pattern.",
        rdfs:label "remodelled Part 2"


AnnotationProperty: lci:seeAlsoPart2

    Annotations:
        rdfs:label "see also Part 2",
        rdfs:comment "Annotation property for recording annotations of entity types given in the original
            →Part 2 of ISO 15926, where the Part 12 entity type is related to the Part 2 entity type, so
            →that the definition of the latter may be a useful reference."


AnnotationProperty: lci:templateReference

    Annotations:
        rdfs:comment "Superrelation for referring to ontology resources that act as parameter values in
            →\"template\" short-cut relations that represent complex patterns. These are informal with
            →regard to the OWL semantics.",
        rdfs:label "template reference"


AnnotationProperty: lci:tplQuality

    Annotations:
        rdfs:label "Quality parameter",
        rdfs:comment "Annotation for templates that have a Quality relation role. Annotations should point
            →to subproperties of hasQuality; for example, \"has mass\" for a generic mass assignment, or
            →\"body temperature\"."

    SubPropertyOf:
        lci:templateReference

    Range:
        <http://www.w3.org/2002/07/owl#ObjectProperty>


AnnotationProperty: lci:tplQuantification

    Annotations:
        rdfs:comment "Annotation for templates that have a Quantification role. Annotations should point to
            →subproperties of qualityQuantifiedAs; for example, \"has mass\" for a generic mass assignment,
            →or \"body temperature\".",
        rdfs:label "Quantification parameter"

    SubPropertyOf:
```

```
        lci:templateReference

    Range:
        <http://www.w3.org/2002/07/owl#ObjectProperty>


AnnotationProperty: lci:tplUOM

    Annotations:
        rdfs:label "UOM parameter",
        rdfs:comment "Annotation for templates that have a Unit of Measure role. Annotations should point to
            →individual members of lci:UnitOfMeasure."

    SubPropertyOf:
        lci:templateReference

    Range:
        <http://standards.iso.org/iso/15926/UnitOfMeasure>


AnnotationProperty: owl:versionInfo


AnnotationProperty: rdfs:comment


AnnotationProperty: rdfs:label


AnnotationProperty: rdfs:seeAlso


AnnotationProperty: skos:definition

    Annotations:
        rdfs:comment "This SKOS relation replaces the Part 2 \"definition\" attribute.",
        rdfs:label "definition"


AnnotationProperty: skos:example

    Annotations:
        rdfs:comment "This SKOS relation replaces the Part 2 \"example\" attribute.",
        rdfs:label "example"


AnnotationProperty: skos:scopeNote

    Annotations:
        rdfs:label "scopeNote",
        rdfs:comment "This SKOS relation replaces the Part 2 \"definition\" attribute."


Datatype: rdf:PlainLiteral


Datatype: rdfs:Literal


Datatype: xsd:dateTime
```

```
Datatype: xsd:decimal


Datatype: xsd:float


Datatype: xsd:integer


Datatype: xsd:string


ObjectProperty: lci:after

    Annotations:
        rdfs:label "after",
        lci:CD "The domain and range of this relation is restricted to activities for the DL profile, where
            →no such restriction is included in the CD.",
        rdfs:comment "Use this relation to state that one activity after before another."

    SubPropertyOf:
        lci:occursRelativeTo


ObjectProperty: lci:approvedBy

    Annotations:
        lci:seeAlsoPart2 "ClassOfApprovalByStatus: EXAMPLE approved, approved with comments, disapproved
            →with comments are examples of [class_of_approval_by_status].",
        lci:remodelledPart2 "Approval: EXAMPLE The [involvement_by_reference] of a plant design with a
            →construction activity, being approved by the site manager, is an example of an [approval].",
        lci:remodelledPart2 "Approval: NOTE Care should be taken as to what is approved. Sometimes it will
            →not be say a pump that is approved, but the participation of the pump in a particular
            →[activity], or member of some [class_of_activity].",
        lci:remodelledPart2 "ClassOfApproval: EXAMPLE That site managers approve design specifications for
            →construction (a [class_of_involvement_by_reference]) is an example of [class_of_approval].",
        lci:remodelledPart2 "Approval: An [approval] is a [relationship] that indicates that a
            →[relationship] has been approved by a [possible_individual] that is an approver.",
        rdfs:comment "Relation for stating that some item or activity was approved by an entity, typically a
            →person or an organisation.",
        lci:seeAlsoPart2 "ClassOfApprovalByStatus: A [class_of_approval_by_status] is a
            →[class_of_relationship] that indicates a status of the approval that is independent of what is
            →being approved by whom.",
        lci:remodelledPart2 "ClassOfApproval: A [class_of_approval] is a [class_of_relationship] whose
            →members are members of [approval] that indicates that members of the [class_of_individual] are
            →approvers in an [approval] for the members of the [class] that are approved.",
        rdfs:label "approvedBy"

    SubPropertyOf:
        lci:interests


ObjectProperty: lci:arrangedPartOf

    Annotations:
        rdfs:label "arrangedPartOf"

    SubPropertyOf:
        lci:partOf

    InverseOf:
        lci:hasArrangedPart
```

```
ObjectProperty: lci:assembledPartOf

    Annotations:
        rdfs:label "assembledPartOf"

    SubPropertyOf:
        lci:arrangedPartOf

    InverseOf:
        lci:hasAssembledPart


ObjectProperty: lci:before

    Annotations:
        rdfs:label "before",
        lci:CD "The domain and range of this relation is restricted to activities for the DL profile, where
            →no such restriction is included in the CD.",
        rdfs:comment "Use this relation to state that one activity occurs before another."

    SubPropertyOf:
        lci:occursRelativeTo


ObjectProperty: lci:begins

    Annotations:
        lci:seeAlsoPart2 "ClassOfCauseOfBeginningOfClassOfIndividual: A
            →[class_of_cause_of_beginning_of_class_of_individual] is a [class_of_relationship] that
            →indicates that a member of a [class_of_activity] causes the beginning of a member of a
            →[class_of_individual].",
        lci:remodelledPart2 "Beginning: A [beginning] is a [temporal_bounding] that marks the temporal start
            →of a [possible_individual].",
        rdfs:label "begins",
        lci:remodelledPart2 "Beginning: EXAMPLE 1 The relation that indicates that the [point_in_time] known
            →as 0000hrs 1st July 1999 UTC is the beginning of the [period_in_time] known as July 1999 UTC
            →can be represented by an instance of [beginning].
EXAMPLE 2 The relation that indicates that the [event] 'loading complete' marks the start of the
    →[possible_individual] 'loading plant idle' can be represented by an instance of [beginning].",
        lci:seeAlsoPart2 "ClassOfCauseOfBeginningOfClassOfIndividual: EXAMPLE A car manufacturing activity
            →causes the beginning of a car."

    SubPropertyOf:
        lci:temporalBoundOf

    InverseOf:
        lci:hasBeginning


ObjectProperty: lci:causes

    Annotations:
        rdfs:label "causes",
        lci:remodelledPart2 "CauseOfEvent: A [cause_of_event] is a [relationship] that indicates that the
            →caused [event] is caused by the causer [activity].",
        lci:remodelledPart2 "CauseOfEvent: EXAMPLE The relation that indicates that the tanker loading
            →activity caused the [event] described as 'tank liquid level full' can be represented by an
            →instance of [cause_of_event].",
        lci:CD "The CD has no domain or range restrictions, but mentions Event in the description. For the
            →DL profile, we lift the restriction to allow also non-instantaneous events to stand in
```

```
        →\"causes\" relationships. We also make \"causes\" a subrelation of \"before\"."

    SubPropertyOf:
        lci:before


ObjectProperty: lci:connectedTo

    Annotations:
        rdfs:label "connectedTo",
        lci:CD "For the DL profile, we add the restriction that only physical objects may be connected. We
            →also add a symmetry constraint.",
        lci:equivalentPart2 "ClassOfConnectionOfIndividual: EXAMPLE Electrical connection between wires is a
            →[class_of_connection_of_individual].",
        lci:equivalentPart2 "ClassOfConnectionOfIndividual: A [class_of_connection_of_individual] is a
            →[class_of_relationship] whose members are members of [connection_of_individual]. It indicates
            →that a member of the class_of_side_1 [class_of_individual] can be connected to a member of the
            →class_of_side_2 [class_of_individual].",
        lci:equivalentPart2 "ClassOfConnectionOfIndividual: NOTE 1 The class_of_side_1 and class_of_side_2
            →indicate the [class_of_individual] that is the side_1 and side_2 respectively in a
            →[connection_of_individual] that is a member of this [class_of_connection_of_individual].
NOTE 2 Flexible, rigid, and welded cannot be represented as instances of
    →[class_of_connection_of_individual], these are classes of the materials connected or used in the
    →connection.",
        lci:equivalentPart2 "ConnectionOfIndividual: A [connection_of_individual] is a [relationship] that
            →indicates that matter, energy, or both can be transferred between the members of
            →[possible_individual] that are connected, either directly or indirectly.",
        lci:equivalentPart2 "ConnectionOfIndividual: NOTE There is no significance to the ordering of the
            →two related instances of [possible_individual]. The names side_1 and side_2 serve only to
            →distinguish the attributes."

    Characteristics:
        Symmetric

    Domain:
        lci:PhysicalObject


ObjectProperty: lci:containedBy

    Annotations:
        lci:equivalentPart2 "ContainmentOfIndividual: NOTE Containment is distinct from composition; in
            →composition the whole consists of all of its part, with containment, what is contained is not
            →a part of the container.",
        lci:equivalentPart2 "ClassOfContainmentOfIndividual: EXAMPLE That 'de-icing fluid' can be contained
            →by a '1500ml screw-top plastic bottle' is a [class_of_containment_of_individual].",
        rdfs:label "containedBy",
        lci:equivalentPart2 "ClassOfContainmentOfIndividual: A [class_of_containment_of_individual] is a
            →[class_of_relative_location] whose members are instances of [containment_of_individual]. It
            →indicates that a member of the class_of_locator [class_of_individual] can contain a member of
            →the class_of_located [class_of_individual].",
        lci:equivalentPart2 "ContainmentOfIndividual: EXAMPLE The contents of a vessel being inside the
            →vessel can be represented by an instance of [containment_of_individual].",
        lci:definitionPart2 "A [containment_of_individual] is a [relative_location] where the located
            →[possible_individual] is contained by the locator [possible_individual] but is not part of
            →it.",
        lci:equivalentPart2 "ContainmentOfIndividual: A [containment_of_individual] is a [relative_location]
            →where the located [possible_individual] is contained by the locator [possible_individual] but
            →is not part of it."

    SubPropertyOf:
        lci:locatedRelativeTo
```

```
    InverseOf:
        lci:contains


ObjectProperty: lci:contains

    Annotations:
        rdfs:label "contains",
        lci:equivalentPart2 "ContainmentOfIndividual: EXAMPLE The contents of a vessel being inside the
            →vessel can be represented by an instance of [containment_of_individual].",
        rdfs:comment "For the DL profile, we restrict this relation to physical objects. Note that this
            →rules out using \"lci:contains\" for spatial locations.",
        lci:equivalentPart2 "ContainmentOfIndividual: A [containment_of_individual] is a [relative_location]
            →where the located [possible_individual] is contained by the locator [possible_individual] but
            →is not part of it.",
        lci:equivalentPart2 "ContainmentOfIndividual: NOTE Containment is distinct from composition; in
            →composition the whole consists of all of its part, with containment, what is contained is not
            →a part of the container."

    SubPropertyOf:
        lci:locatedRelativeTo

    Domain:
        lci:PhysicalObject

    Range:
        lci:PhysicalObject

    InverseOf:
        lci:containedBy


ObjectProperty: lci:creates

    Annotations:
        rdfs:comment "Use this relation to express that an activity brings a physical object into being.
            →(Derived from class_of_cause_of_beginning_of_class_of_individual).",
        lci:CD "This relation is not included in the CD. The CD however has \"causesBeginningOf\" with
            →apparently the same meaning -- bringing about the \"beginning\" of an individual. For the DL
            →profile, we keep the name \"creates\" to avoid confusion with beginning/end talk about
            →temporal bounds of activities. We also restrict the range to physical objects, to distinguish
            →this relation from the \"causes\" relation between activities.",
        lci:definitionPart2 "A [class_of_cause_of_beginning_of_class_of_individual] is a
            →[class_of_relationship] that indicates that a member of a [class_of_activity] causes the
            →beginning of a member of a [class_of_individual].",
        rdfs:label "creates"

    Domain:
        lci:Activity

    Range:
        lci:PhysicalObject


ObjectProperty: lci:datumUOM

    Annotations:
        rdfs:comment "Relation (functional) to assign unit of measure to measurement data.",
        rdfs:label "datumUOM"

    Characteristics:
```

```
        Functional

    Domain:
        lci:QuantityDatum

    Range:
        lci:UnitOfMeasure


ObjectProperty: lci:directlyConnectedTo

    Annotations:
        lci:seeAlsoPart2 "IndirectConnection: EXAMPLE The relation that indicates that there is a railway
            →connection between the cities of London and Paris can be represented by an instance of
            →[indirect_connection].",
        lci:seeAlsoPart2 "IndirectConnection: An [indirect_connection] is a [connection_of_individual] that
            →indicates that side_1 and side_2 are connected via other individuals.",
        lci:equivalentPart2 "DirectConnection: EXAMPLE The relation that indicates that the plug terminating
            →a serial communications cable is connected to the socket on a piece of computer equipment can
            →be represented by an instance of [direct_connection].",
        rdfs:label "directlyConnectedTo",
        lci:equivalentPart2 "DirectConnection: A [direct_connection] is a [connection_of_individual] that
            →indicates that the side_1 and side_2 are directly connected via a common spatial boundary.",
        lci:seeAlsoPart2 "ClassOfIndirectConnection: A [class_of_indirect_connection] is a
            →[class_of_connection_of_individual] whose members are members of [indirect_connection].",
        lci:CD "For the DL profile, we leave out the CD relation \"indirectlyConnectedTo\".",
        lci:equivalentPart2 "ClassOfDirectConnection: EXAMPLE Three-pin electrical plug into three-pin
            →socket is an example of [class_of_direct_connection].",
        lci:equivalentPart2 "ClassOfDirectConnection: A [class_of_direct_connection] is a
            →[class_of_connection_of_individual] whose members are members of [direct_connection].",
        lci:seeAlsoPart2 "ClassOfIndirectConnection: EXAMPLE Drip pipe indirectly connected to drain funnel
            →is an example of [class_of_indirect_connection]."

    SubPropertyOf:
        lci:connectedTo


ObjectProperty: lci:ends

    Annotations:
        lci:equivalentPart2 "Ending: An [ending] is a [temporal_bounding] that marks the end of a
            →[possible_individual].",
        lci:seeAlsoPart2 "ClassOfCauseOfEndingOfClassOfIndividual: EXAMPLE A car crushing activity causes
            →the end of the life of a car.",
        rdfs:label "ends",
        lci:equivalentPart2 "Ending: EXAMPLE 1 The relation that indicates that the [point_in_time] known as
            →0000hrs 1st July 1999 GMT is the end of the [period_in_time] known as June 1999 GMT can be
            →represented by an instance of [ending].
EXAMPLE 2 The relation that indicates that the [event] 'loading complete' marks the end of the
    →[possible_individual] 'loading plant operating period 1' (a temporal part of the loading plant) is an
    →instance of [ending].",
        lci:seeAlsoPart2 "ClassOfCauseOfEndingOfClassOfIndividual: A
            →[class_of_cause_of_ending_of_class_of_individual] is a [class_of_relationship] that indicates
            →that a member of the [class_of_activity] causes the ending of a member of the
            →[class_of_individual]."

    SubPropertyOf:
        lci:temporalBoundOf

    InverseOf:
        lci:hasEnd
```

```
ObjectProperty: lci:featureOf

    Annotations:
        lci:equivalentPart2 "FeatureWholePart: NOTE This includes wholes that cannot be non-destructively
            →disassembled and reassembled such as the cast inlet flange of a pump.",
        rdfs:label "featureOf",
        lci:equivalentPart2 "ClassOfFeatureWholePart: A [class_of_feature_whole_part] is a
            →[class_of_arrangement_of_individual] whose members are instances of [feature_whole_part].",
        lci:equivalentPart2 "FeatureWholePart: A [feature_whole_part] is an [arrangement_of_individual] that
            →indicates that the part is a non-separable, contiguous part of the whole.",
        lci:equivalentPart2 "ClassOfFeatureWholePart: EXAMPLE Thermowells have stems, and tables have tops
            →are examples of [class_of_feature_whole_part].",
        lci:equivalentPart2 "FeatureWholePart: EXAMPLE The relation that indicates that a flange face is
            →part of a flange can be represented by an instance of [feature_whole_part]."

    SubPropertyOf:
        lci:arrangedPartOf

    InverseOf:
        lci:hasFeature


ObjectProperty: lci:hasArrangedPart

    Annotations:
        lci:remodelledPart2 "ArrangementOfIndividual: EXAMPLE 1 The relationship that indicates that a
            →particular aircraft is flying as part of a formation can be represented by an instance of
            →[arrangement_of_individual].
EXAMPLE 2 The relationship that indicates that a particular bin in a warehouse is part of the warehouse
    →layout can be represented by an instance of [arrangement_of_individual].",
        rdfs:label "hasArrangedPart",
        lci:remodelledPart2 "ClassOfArrangedIndividual: A [class_of_arranged_individual] is a
            →[class_of_individual] whose members have a distinct form that may arise from the arrangement
            →of their parts.",
        lci:remodelledPart2 "ArrangementOfIndividual: NOTE 1 The term \"arranged\" implies that parts have
            →particular roles with respect to the whole.
NOTE 2 The natures of the relations to other parts of the whole are not specified by the arrangement
    →relation. Relationships like [connection_of_individual] and [relative_location] would indicate this.",
        lci:definitionPart2 "An [arrangement_of_individual] is a [composition_of_individual] that indicates
            →that the part is a part of an [arranged_individual]. The temporal extent of the part is that
            →of the whole. An [arrangement_of_individual] may be an [assembly_of_individual].",
        lci:remodelledPart2 "ArrangementOfIndividual: An [arrangement_of_individual] is a
            →[composition_of_individual] that indicates that the part is a part of an
            →[arranged_individual]. The temporal extent of the part is that of the whole. An
            →[arrangement_of_individual] may be an [assembly_of_individual]. ",
        lci:remodelledPart2 "ClassOfArrangementOfIndividual: EXAMPLE The fact that water is made up of H2O
            →molecules is an instance of [class_of_arrangement_of_individual].",
        lci:remodelledPart2 "ClassOfArrangedIndividual: NOTE 1 The ONEOF constraint on some of the subtypes
            →does not prevent a particular [possible_individual] from being, say, a member of a particular
            →[arranged_individual] classified by [class_of_biological_matter] and a member of a particular
            →[class_of_composite_material]. It is only the classes themselves that are not members of more
            →than one of the entity types.
NOTE 2 Specifications or descriptions of useful objects are often intersections of several arrangement
    →classes, allowing both shape and material aspects to be constrained. In this part of ISO 15926, such
    →intersections are members of [class_of_arranged_individual], [class_of_feature],
    →[class_of_inanimate_physical_object], [class_of_organization], [class_of_activity],
    →[class_of_organism], or [class_of_information_object].",
        lci:remodelledPart2 "ArrangedIndividual: An [arranged_individual] is a [possible_individual] that
            →has parts that play distinct roles with respect to the whole. The qualities of an
            →[arranged_individual] are distinct from the qualities of its parts.",
```

```
        lci:remodelledPart2 "ClassOfArrangedIndividual: EXAMPLE Robocop is a [class_of_arranged_individual]
            →that has some parts that are members of some [class_of_inanimate_physical_object] and parts
            →that are members of some [class_of_organism].",
        lci:remodelledPart2 "ClassOfArrangementOfIndividual: A [class_of_arrangement_of_individual] is a
            →[class_of_composition_of_individual] whose members are instances of
            →[arrangement_of_individual].",
        rdfs:comment "In line with intended use, for the DL profile this relation has a domain restricted to
            →physical objects.",
        lci:remodelledPart2 "ArrangedIndividual: EXAMPLE 1 The vessel with serial number V-1234 is an
            →[arranged_individual].
EXAMPLE 2 The company Bloggs & Co. is an [arranged_individual].
EXAMPLE 3 A laptop computer that consists of the main unit with its removable CD-ROM and floppy disk drives
    →and power supply cables is an [arranged_individual]."

    SubPropertyOf:
        lci:hasPart

    Domain:
        lci:PhysicalObject

    InverseOf:
        lci:arrangedPartOf


ObjectProperty: lci:hasAssembledPart

    Annotations:
        lci:remodelledPart2 "AssemblyOfIndividual: NOTE Composition of molecules and smaller is represented
            →through instances of [class_of_arrangement_of_individual].",
        lci:definitionPart2 "An [assembly_of_individual] is an [arrangement_of_individual] that indicates
            →that the part is connected directly or indirectly to other parts of the whole. The parts and
            →wholes are super-molecular objects.",
        lci:remodelledPart2 "AssemblyOfIndividual: An [assembly_of_individual] is an
            →[arrangement_of_individual] that indicates that the part is connected directly or indirectly
            →to other parts of the whole. The parts and wholes are super-molecular objects.",
        lci:remodelledPart2 "ClassOfAssemblyOfIndividual: EXAMPLE That impellers are parts of centrifugal
            →pumps is a [class_of_assembly_of_individual].",
        rdfs:label "hasAssembledPart",
        lci:remodelledPart2 "AssemblyOfIndividual: EXAMPLE The relation that indicates that a temporal part
            →of an impeller is a part of an assembled pump can be represented by an instance of
            →[assembly_of_individual].",
        rdfs:comment "This is the recommended (super-) relation for capturing physical breakdown of
            →mechanical assemblies.",
        lci:remodelledPart2 "ClassOfAssemblyOfIndividual: A [class_of_assembly_of_individual] is a
            →[class_of_arrangement_of_individual] whose members are instances of [assembly_of_individual]."

    SubPropertyOf:
        lci:hasArrangedPart

    InverseOf:
        lci:assembledPartOf


ObjectProperty: lci:hasBeginning

    Annotations:
        lci:remodelledPart2 "Beginning: A [beginning] is a [temporal_bounding] that marks the temporal start
            →of a [possible_individual].",
        rdfs:label "hasBeginning",
        lci:remodelledPart2 "Beginning: EXAMPLE 1 The relation that indicates that the [point_in_time] known
            →as 0000hrs 1st July 1999 UTC is the beginning of the [period_in_time] known as July 1999 UTC
            →can be represented by an instance of [beginning].
```

```
EXAMPLE 2 The relation that indicates that the [event] 'loading complete' marks the start of the
    →[possible_individual] 'loading plant idle' can be represented by an instance of [beginning]."

    SubPropertyOf:
        lci:hasTemporalBound

    InverseOf:
        lci:begins


ObjectProperty: lci:hasEnd

    Annotations:
        lci:equivalentPart2 "Ending: An [ending] is a [temporal_bounding] that marks the end of a
            →[possible_individual].",
        rdfs:label "hasEnd",
        lci:equivalentPart2 "Ending: EXAMPLE 1 The relation that indicates that the [point_in_time] known as
            →0000hrs 1st July 1999 GMT is the end of the [period_in_time] known as June 1999 GMT can be
            →represented by an instance of [ending].
EXAMPLE 2 The relation that indicates that the [event] 'loading complete' marks the end of the
    →[possible_individual] 'loading plant operating period 1' (a temporal part of the loading plant) is an
    →instance of [ending]."

    SubPropertyOf:
        lci:hasTemporalBound

    InverseOf:
        lci:ends


ObjectProperty: lci:hasFeature

    Annotations:
        rdfs:comment "Example of usage: stating that an entity has a surface suitable for connection, such
            →as a flange face.",
        lci:definitionPart2 "A [class_of_feature] is a [class_of_arranged_individual] whose members are
            →contiguous, non-separable parts of some [possible_individual] and have an incompletely defined
            →boundary.",
        lci:equivalentPart2 "FeatureWholePart: NOTE This includes wholes that cannot be non-destructively
            →disassembled and reassembled such as the cast inlet flange of a pump.",
        rdfs:label "hasFeature",
        lci:equivalentPart2 "FeatureWholePart: A [feature_whole_part] is an [arrangement_of_individual] that
            →indicates that the part is a non-separable, contiguous part of the whole.",
        lci:equivalentPart2 "FeatureWholePart: EXAMPLE The relation that indicates that a flange face is
            →part of a flange can be represented by an instance of [feature_whole_part]."

    SubPropertyOf:
        lci:hasArrangedPart

    InverseOf:
        lci:featureOf


ObjectProperty: lci:hasFunction

    Annotations:
        lci:remodelledPart2 "FunctionalPhysicalObject: 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.",
        rdfs:label "hasFunction",
```

```
        lci:remodelledPart2 "FunctionalPhysicalObject: EXAMPLE The heat exchanger system known as tag
            →E-4507, which is part of a distillate transfer system, can be represented by an instance of
            →[functional_physical_object]. Note that this is distinct from the \"shell and tube heat
            →exchanger manufacture number ES/1234\" that was installed as E-4507 when the plant was first
            →built and later removed when worn out, to be replaced by a new heat exchanger with different
            →serial number. \"Shell and tube heat exchanger manufacture number ES/1234\" and its
            →differently numbered replacement can be represented by instances of
            →[materialized_physical_object]. When ES/1234 is installed as E-4507 there is a temporal part
            →of ES/1234 that is also a temporal part of E-4507.",
        rdfs:comment "Inspired by BFO's \"has function\" (RO_0000085)."

    Range:
        lci:Function


ObjectProperty: lci:hasPart

    Annotations:
        lci:definitionPart2 "A [composition_of_individual] is a [relationship] that indicates that the part
            →[possible_individual] is a part of the whole [possible_individual]. A simple composition is
            →indicated, unless a subtype is instantiated too. [composition_of_individual] is transitive.",
        lci:remodelledPart2 "CompositionOfIndividual: EXAMPLE A grain of sand being part of a pile of sand
            →is an example of [composition_of_individual].",
        lci:remodelledPart2 "ClassOfCompositionOfIndividual: A [class_of_composition_of_individual] is a
            →[class_of_relationship] whose members are members of [composition_of_individual].",
        lci:remodelledPart2 "ClassOfCompositionOfIndividual: EXAMPLE That piles of sand may have grains of
            →sand as parts is an example of [class_of_composition_of_individual].",
        lci:remodelledPart2 "ClassOfClassOfComposition: A [class_of_class_of_composition] is a
            →[class_of_class_of_relationship] whose members are instances of
            →[class_of_composition_of_individual]. It indicates that a member of a member of the
            →class_of_class_of_part is a part of a member of an instance of the class_of_class_of_whole.",
        rdfs:label "hasPart",
        lci:remodelledPart2 "ClassOfClassOfComposition: EXAMPLE Toxicity description is a
            →class_of_class_of_part of a material data sheet, where the description \"has carcinogenic
            →components\" is a class_of_part on the Mogas Material Safety Data Sheet, and copy #5 of the
            →Mogas Material Safety Data Sheet has \"has carcinogenic components\" as a part.",
        lci:remodelledPart2 "CompositionOfIndividual: A [composition_of_individual] is a [relationship] that
            →indicates that the part [possible_individual] is a part of the whole [possible_individual]. A
            →simple composition is indicated, unless a subtype is instantiated too.
            →[composition_of_individual] is transitive. ",
        lci:remodelledPart2 "CompositionOfIndividual: NOTE Simple composition means that for example no
            →arrangement of parts is necessarily implied or of concern. Where there is an arrangement of
            →parts, this is indicated by an [arrangement_of_individual], which, by being a subtype, implies
            →also a simple composition."

    InverseOf:
        lci:partOf


ObjectProperty: lci:hasParticipant

    Annotations:
        lci:definitionPart2 "A [participation] is a [composition_of_individual] that indicates that a
            →[possible_individual] is a participant in an [activity].",
        rdfs:comment "This is the recommended superrelation for roles that entities can take in activities
            →-- the agent, the matter being acted upon, etc. (There may be reason to include lci:creates as
            →a subrelation of this relation.)",
        lci:equivalentPart2 "Participation: NOTE The [possible_individual] that is the part in the
            →[participation] is may be a temporal part of a [whole_life_individual] that is classified by
            →the [role_and_domain] that indicates the role it plays in the [activity].",
        rdfs:comment "Note that BFO does not have 'has participant' as a subrelation of 'has part'. This can
            →be motivated in that 4D objects are not obviously able to have 3D parts.",
```

```
        lci:equivalentPart2 "Participation: A [participation] is a [composition_of_individual] that
            →indicates that a [possible_individual] is a participant in an [activity].",
        lci:equivalentPart2 "Participation: EXAMPLE The relationship between the temporal part of P1234 that
            →performs the discharge of the Motor Vessel Murex on 2nd December 2002, and the activity that
            →is that discharge of that vessel is a [participation].",
        rdfs:label "hasParticipant"

    SubPropertyOf:
        lci:hasPart

    Domain:
        lci:Activity

    InverseOf:
        lci:participantIn


ObjectProperty: lci:hasQuality

    Annotations:
        rdfs:label "hasQuality",
        lci:CD "With the CD version, qualities (in the CD, called quantities or properties) are assigned by
            →way of classification. This relation to individual qualities is only implicit in Part 2."

    Range:
        lci:Quality


ObjectProperty: lci:hasRole

    Annotations:
        rdfs:label "hasRole",
        rdfs:comment "Inspired by BFO's \"has role\" (RO_0000087)"

    Range:
        lci:Role

    InverseOf:
        lci:roleOf


ObjectProperty: lci:hasTemporalBound

    Annotations:
        lci:remodelledPart2 "TemporalBounding: A [temporal_bounding] is a [composition_of_individual] that
            →indicates that the part [event] is a temporal boundary of the whole [possible_individual].",
        rdfs:label "hasTemporalBound"

    SubPropertyOf:
        lci:hasTemporalPart

    InverseOf:
        lci:temporalBoundOf


ObjectProperty: lci:hasTemporalPart

    Annotations:
        lci:equivalentPart2 "ClassOfTemporalWholePart: EXAMPLE The class that indicates that Crude
            →Distillation Units may have a maximum naphtha mode can be represented by an instance of
            →[class_of_temporal_whole_part].",
```

```
        lci:equivalentPart2 "TemporalWholePart: EXAMPLE 1 The relation that indicates that an operating
            →period of a pump is a temporal part of the pump can be represented by an instance of
            →[temporal_whole_part].
EXAMPLE 2 The relationship that indicates that the time period known as March 1999 is part of the period
    →known as 1st Quarter 1999 can be represented by an instance of [temporal_whole_part].",
        lci:equivalentPart2 "ClassOfTemporalWholePart: A [class_of_temporal_whole_part] is a
            →[class_of_composition_of_individual] whose members are members of [temporal_whole_part].",
        lci:equivalentPart2 "TemporalWholePart: A [temporal_whole_part] is a [composition_of_individual]
            →that indicates that one [possible_individual] is a temporal part of another
            →[possible_individual]. The spatial extent of the temporal part is that of the temporal whole
            →for the period of the existence of the temporal part. Relationships that apply to the whole
            →[possible_individual] also apply to the temporal parts of the [possible_individual], except
            →when the relationships relate to the temporal nature of the whole. So if a
            →[possible_individual] is connected so are all its temporal parts, but being a
            →[whole_life_individual] is not inherited by its temporal parts.",
        lci:CD "The DL profile restricts temporal parts to Activity individuals.",
        rdfs:label "hasTemporalPart",
        lci:equivalentPart2 "TemporalWholePart: NOTE Since [temporal_whole_part] is transitive (inherited
            →from its supertype) a hierarchy of temporal parts is possible, with a [whole_life_individual]
            →at the top."

    SubPropertyOf:
        lci:hasPart

    Domain:
        lci:Activity

    InverseOf:
        lci:temporalPartOf


ObjectProperty: lci:interests

    Annotations:
        rdfs:label "interests",
        rdfs:comment "Derived from \"LifecycleStage\" of Part 2, this is a superproperty suitable for
            →various intentional relationships, such as planning, approving, or ordering. The Part 2 name
            →\"lifecycle stage\" is likely to confuse, but the intended use of this type is clear enough
            →from this Part 2 annotation to ClassOfLifecycleStage: \"EXAMPLE Planned, required, expected,
            →and proposed can be represented by instances of [class_of_lifecycle_stage].\"",
        lci:remodelledPart2 "ClassOfLifecycleStage: EXAMPLE Planned, required, expected, and proposed can be
            →represented by instances of [class_of_lifecycle_stage].",
        lci:remodelledPart2 "ClassOfLifecycleStage: A [class_of_lifecycle_stage] is a
            →[class_of_relationship] whose members are members of [lifecycle_stage].",
        lci:remodelledPart2 "LifecycleStage: EXAMPLE The relation that links a possible building to a
            →temporal part of the XYZ Corp. can be represented by an instance of [lifecycle_stage]. The
            →nature of that [lifecycle_stage] (e.g. 'planned') can be expressed by classifying with the
            →applicable [class_of_lifecycle_stage].",
        lci:remodelledPart2 "LifecycleStage: A [lifecycle_stage] is a [relationship] that indicates the
            →interest that a [possible_individual] has in some [possible_individual]. "


ObjectProperty: lci:locatedRelativeTo

    Annotations:
        lci:remodelledPart2 "RelativeLocation: EXAMPLE A being the located relative to B being the locator
            →in a [relative_location] that is classified by the [class_of_relative_location] above,
            →indicates that A is above B.",
        lci:remodelledPart2 "RelativeLocation: A [relative_location] is a [relationship] that indicates that
            →the position of one [possible_individual] is relative to another.",
        lci:remodelledPart2 "ClassOfRelativeLocation: A [class_of_relative_location] is a
            →[class_of_relationship] whose members are instances of [relative_location].",
```

```
        lci:remodelledPart2 "RelativeLocation: NOTE The [classification] of the [relative_location]
            →indicates the nature of the [relative_location], e.g. above, below, beside.",
        lci:remodelledPart2 "ClassOfRelativeLocation: EXAMPLE Beside, above, and below are examples of
            →[class_of_relative_location].",
        rdfs:label "locatedRelativeTo",
        lci:definitionPart2 "A [relative_location] is a [relationship] that indicates that the position of
            →one [possible_individual] is relative to another."


ObjectProperty: lci:occursRelativeTo

    Annotations:
        rdfs:label "occursRelativeTo",
        lci:remodelledPart2 "ClassOfTemporalSequence: EXAMPLE 1 The link that indicates that members of July
            →follow members of June can be represented by an instance of [class_of_temporal_sequence].
EXAMPLE 2 The link that indicates that emptying activities for a tank precede cleaning activities can be
    →represented by an instance of [class_of_temporal_sequence].",
        lci:remodelledPart2 "TemporalSequence: EXAMPLE 1 The [relationship] that indicates that the
            →[possible_individual] that is the construction phase of a plant precedes the
            →[possible_individual] that is the commissioning phase of a plant can be represented by an
            →instance of [temporal_sequence].
EXAMPLE 2 The [relationship] that indicates that the [period_in_time] known as the industrial revolution
    →preceded the [period_in_time] known as the information revolution can be represented by an instance
    →of [temporal_sequence].",
        lci:remodelledPart2 "TemporalSequence: A [temporal_sequence] is a [relationship] that indicates that
            →one [possible_individual] precedes another in a temporal sense.",
        rdfs:comment "This relation is introduced for the DL profile as a top relation for various temporal
            →relations between activities.",
        lci:remodelledPart2 "ClassOfTemporalSequence: A [class_of_temporal_sequence] is a
            →[class_of_relationship] where the sequence is of a temporal nature."

    Domain:
        lci:Activity

    Range:
        lci:Activity


ObjectProperty: lci:partOf

    Annotations:
        rdfs:label "partOf"

    InverseOf:
        lci:hasPart


ObjectProperty: lci:participantIn

    Annotations:
        lci:equivalentPart2 "Participation: NOTE The [possible_individual] that is the part in the
            →[participation] is may be a temporal part of a [whole_life_individual] that is classified by
            →the [role_and_domain] that indicates the role it plays in the [activity].",
        lci:equivalentPart2 "ClassOfParticipation: A [class_of_participation] is a
            →[class_of_composition_of_individual] that indicates a member of an instance of
            →[participating_role_and_domain] participates in a member of an instance of
            →[class_of_activity].",
        lci:equivalentPart2 "ClassOfParticipation: EXAMPLE \"Conductor of a musical performance\" is an
            →example of [class_of_participation].",
        lci:equivalentPart2 "Participation: EXAMPLE The relationship between the temporal part of P1234 that
            →performs the discharge of the Motor Vessel Murex on 2nd December 2002, and the activity that
            →is that discharge of that vessel is a [participation].",
```

```
        lci:equivalentPart2 "Participation: A [participation] is a [composition_of_individual] that
            →indicates that a [possible_individual] is a participant in an [activity].",
        rdfs:label "participantIn"

    SubPropertyOf:
        lci:partOf

    InverseOf:
        lci:hasParticipant


ObjectProperty: lci:qualityQuantifiedAs

    Annotations:
        lci:equivalentPart2 "PropertyQuantification: EXAMPLE The link that maps a particular mass to the
            →number 4.2 can be represented by an instance of [property_quantification].",
        lci:equivalentPart2 "PropertyQuantification: NOTE 1 The actual representation of the number is done
            →by linking the [arithmetic_number] to a [class_of_EXPRESS_information_representation] via a
            →[class_of_representation_of_thing].
NOTE 2 The unit or scale of the quantification is given by classifying the [property_quantification] by a
    →[scale].",
        rdfs:comment "This relation is inspired by the relation \"is quality measured as\" of the
            →Information Artefact Ontology. The term \"quantified\" replaces \"measured\" to support cases
            →where measurement is not involved, as in e.g. estimates.",
        rdfs:label "qualityQuantifiedAs",
        lci:equivalentPart2 "PropertyQuantification: A [property_quantification] is a [functional_mapping]
            →whose members map a [property] to an [arithmetic_number]."

    SubPropertyOf:
        lci:representedBy

    Domain:
        lci:Quality

    Range:
        lci:QuantityDatum


ObjectProperty: lci:realizedIn

    Annotations:
        rdfs:label "realizedIn",
        rdfs:comment "Inspired by BFO's \"realized in\" (BFO_0000054)"


ObjectProperty: lci:representedBy

    Annotations:
        lci:seeAlsoPart2 "ClassOfClassOfInformationRepresentation: A
            →[class_of_class_of_information_representation] is a [class_of_class_of_individual] that
            →classifies information representation classes.",
        lci:equivalentPart2 "RepresentationOfThing: A [representation_of_thing] is a [relationship] that
            →indicates that a [possible_individual] is a sign for a [thing].",
        rdfs:comment "Following Part 2, this is the top level relation from information objects to things.",
        lci:remodelledPart2 "ClassOfInformationRepresentation: EXAMPLE The texts formed with the pattern of
            →characters 's' concatenated with 'u' concatenated with 'n' are members of the 'sun'
            →[class_of_information_representation].",
        lci:seeAlsoPart2 "ClassOfClassOfRepresentation: A [class_of_class_of_representation] is a
            →[class_of_class_of_relationship] whose members are instances of
            →[class_of_representation_of_thing].",
        lci:seeAlsoPart2 "ClassOfClassOfInformationRepresentation: EXAMPLE Integer Octal is a
            →[class_of_class_of_information_representation] whose members are all the information
```

```
        →representation classes that correspond to Octal formatted integers.",
    lci:equivalentPart2 "RepresentationOfThing: EXAMPLE The relationship between a nameplate with its
        →serial number and other data, and a particular pressure vessel
        →([materialized_physical_object]) is an example of [representation_of_thing] that is an
        →[identification].",
    rdfs:label "representedBy",
    lci:remodelledPart2 "ClassOfInformationPresentation: A [class_of_information_presentation] is a
        →[class_of_arranged_individual] that distinguishes styles for presenting information. ",
    lci:seeAlsoPart2 "ClassOfClassOfRepresentation: EXAMPLE The link that indicates that members of the
        →class 'document' can be represented by patterns of the class 'XML' is a
        →[class_of_class_of_representation].",
    rdfs:seeAlso "Also see \"is about\" IAO_0000136 of the Information Artifact Ontology, which is
        →probably better named for a maximally general relation of \"aboutness\" (but note that \"is
        →about\" goes in the opposite direction of \"representedIn\").",
    lci:remodelledPart2 "ClassOfInformationRepresentation: A [class_of_information_representation] is a
        →[class_of_arranged_individual] that defines a pattern that represents information.",
    lci:equivalentPart2 "RepresentationOfThing: NOTE In general it will be
        →[class_of_representation_of_thing] that will be of interest, rather than each
        →[representation_of_thing]. However, [representation_of_thing] will be of interest when
        →individual copies of documents are managed and controlled.",
    lci:remodelledPart2 "ClassOfInformationPresentation: EXAMPLE The character styles bold, italic,
        →Times New Roman, and 16pt can be represented as instances of
        →[class_of_information_presentation]."

  Range:
    lci:InformationObject


ObjectProperty: lci:roleOf

  Annotations:
    rdfs:comment "Inspired by BFO's \"role of\" (RO_0000081)",
    rdfs:label "roleOf"

  InverseOf:
    lci:hasRole


ObjectProperty: lci:temporalBoundOf

  Annotations:
    rdfs:label "temporalBoundOf"

  SubPropertyOf:
    lci:temporalPartOf

  InverseOf:
    lci:hasTemporalBound


ObjectProperty: lci:temporalPartOf

  Annotations:
    rdfs:label "temporalPartOf"

  SubPropertyOf:
    lci:partOf

  InverseOf:
    lci:hasTemporalPart
```

```
ObjectProperty: lci:realizedIn

    Domain:
        lci:Function

    Range:
        lci:Activity


DataProperty: lci:approvedOn

    Annotations:
        rdfs: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.",
        lci:remodelledPart2 "Approval: EXAMPLE The [involvement_by_reference] of a plant design with a
            →construction activity, being approved by the site manager, is an example of an [approval].",
        lci:remodelledPart2 "Approval: NOTE Care should be taken as to what is approved. Sometimes it will
            →not be say a pump that is approved, but the participation of the pump in a particular
            →[activity], or member of some [class_of_activity].",
        lci:remodelledPart2 "Approval: An [approval] is a [relationship] that indicates that a
            →[relationship] has been approved by a [possible_individual] that is an approver.",
        rdfs:label "approvedOn"


DataProperty: lci:datumValue

    Annotations:
        rdfs:comment "Consider whether xsd:float is the correct data type. Perhaps it's too limiting to
            →require this type.",
        rdfs:comment "This relation is inspired by the relation \"has measurement value\" of the Information
            →Artefact Ontology.",
        rdfs:label "datumValue"

    Characteristics:
        Functional

    Range:
        xsd:float


DataProperty: lci:qualityQuantityValue

    Annotations:
        rdfs:label "qualityQuantityValue",
        rdfs:comment "This is a super-property for \"template\" relations that combine a quality, the weak
            →lci:qualityQuantifiedAs, and a unit of measure into a simple data property. For instance,
            →\"mass in kilograms\" can be introduced as such a data property, for expressing the mass of an
            →entity on the kilogram scale. lci:qualityQuantifiedAs is \"weak\" in the sense that it doesn't
            →distinguish between designed or estimated, and measured, values."


Class: lci:Activity

    Annotations:
        lci:equivalentPart2 "Activity: An [activity] is a [possible_individual] that brings about change by
            →causing the [event] that marks the [beginning], or the [event] that marks the [ending] of a
            →[possible_individual]. An activity consists of the temporal parts of those members of
            →[possible_individual] that participate in the activity. The participating temporal parts will
            →be classified by the [participating_role_and_domain] that indicates the role of the temporal
            →part in the [activity].",
```

```
        lci:equivalentPart2 "ClassOfActivity: NOTE Behaviour is a term used to describe a
            →[class_of_activity] either where there are preconditions and the [class_of_activity] is a
            →response to those preconditions, e.g. reaction to touching a hot surface, or where the way an
            →activity occurs is described by some property or function, e.g. fluid flow being described by
            →the viscosity of the fluid.",
        rdfs:label "Activity",
        lci:equivalentPart2 "Activity: EXAMPLE Pumping a fluid with a mechanical pump can be represented by
            →an instance of [activity].",
        lci:equivalentPart2 "ClassOfActivity: A [class_of_activity] is a [class_of_arranged_individual]
            →whose members are instances of [activity].",
        lci:equivalentPart2 "ClassOfActivity: EXAMPLE Drilling, distilling, and approving can be represented
            →by instances of [class_of_activity]."


Class: lci:Compound

    Annotations:
        lci:equivalentPart2 "ClassOfCompound: NOTE Compound is being used here in a more general sense than
            →chemical compound.",
        lci:equivalentPart2 "ClassOfCompound: A [class_of_compound] is a [class_of_arranged_individual]
            →whose members consist of arrangements of molecules of the same or different types, bound
            →together by intermolecular forces. This includes both mixtures and alloys.",
        rdfs:label "Compound",
        lci:equivalentPart2 "ClassOfCompound: EXAMPLE Water, sulphuric acid, sand, limestone, and steel can
            →be represented by instances of [class_of_compound]."

    SubClassOf:
        lci:PhysicalObject


Class: lci:Event

    Annotations:
        lci:equivalentPart2 "ClassOfEvent: A [class_of_event] is a [class_of_individual] whose members are
            →members of [event].",
        lci:equivalentPart2 "Event: EXAMPLE The connection of power to a pump is an event that marks the
            →beginning of a temporal part of that pump.",
        lci:equivalentPart2 "ClassOfEvent: EXAMPLE Continuous and instantaneous are instances of
            →[class_of_event]. A continuous event is one such as a stream boundary flowing through a pipe.",
        lci:equivalentPart2 "Event: An [event] is a [possible_individual] with zero extent in time at any
            →point in space-time - a four dimensional plane. An [event] may be at one-time only, or may
            →extend in time at different places, or a combination of both. An [event] is the temporal
            →boundary of one or more [possible_individual]s, although there may be no knowledge of these
            →[possible_individual]s.",
        rdfs:label "Event"

    SubClassOf:
        lci:Activity


Class: lci:Feature

    Annotations:
        lci:equivalentPart2 "ClassOfFeature: A [class_of_feature] is a [class_of_arranged_individual] whose
            →members are contiguous, non-separable parts of some [possible_individual] and have an
            →incompletely defined boundary.",
        rdfs:label "Feature",
        lci:equivalentPart2 "ClassOfFeature: EXAMPLE The classes known as 'mountain', 'groove', 'rim',
            →'nozzle', 'nose', and 'raised face' can all be represented as instances of [class_of_feature]."

    SubClassOf:
        lci:PhysicalObject
```

```
Class: lci:Function

    Annotations:
        rdfs:label "Function",
        lci:equivalentPart2 "ClassOfFunctionalObject: A [class_of_functional_object] is a
            →[class_of_arranged_individual] that indicates the function or purpose of an object.",
        rdfs:comment "Inspired by the BFO class \"function\" (BFO_0000034).",
        lci:equivalentPart2 "ClassOfFunctionalObject: EXAMPLE Pump, valve, and car are examples of
            →[class_of_functional_object]. Particular models of pump, valve, car, etc are instances of
            →[class_of_inanimate_physical_object] that are specializations of these instances of
            →[class_of_functional_object]."


Class: lci:InanimatePhysicalObject

    Annotations:
        lci:equivalentPart2 "ClassOfInanimatePhysicalObject: A [class_of_inanimate_physical_object] is a
            →[class_of_arranged_individual] whose members are not living.",
        rdfs:label "InanimatePhysicalObject",
        lci:equivalentPart2 "ClassOfInanimatePhysicalObject: EXAMPLE The class known as 'oil' can be
            →represented by an instance of [class_of_inanimate_physical_object]."

    SubClassOf:
        lci:PhysicalObject


Class: lci:InformationObject

    Annotations:
        lci:remodelledPart2 "Class: NOTE 1 The membership of a [class] is unchanging as a result of the
            →spatio-temporal paradigm upon which this schema is based. In another paradigm it might be
            →stated that a car is red at one time, and green at another time, indicating that the class of
            →red things and class of green things changed members. However, using a spatio-temporal
            →paradigm, a temporal part, state 1, of the car is red, and another temporal part of the car,
            →state 2, is green. In this way the members of the classes red and green are unchanging. The
            →same principle applies to future temporal parts as to past temporal parts, it is just more
            →likely that the membership of these is not known.
A class may be a member of another class or itself.
NOTE 2 The set theory that applies to classes in this model is non-wellfounded set theory [3]. This permits
    →statements like \"class is a member of class\", unlike traditional set theories such as
    →Zermelo-Fraenkel set theory found in standard texts [4].
There is a null [class] that has no members.
NOTE 3 The known members of a [class] are identified by [classification].
NOTE 4 Although there is only one [class] that has no members, there can be a [class] that has no members
    →in the actual world, but which does have members in other possible worlds.

BIBLIOGRAPHY
[3] ACZEL, Peter. Non-Well-Founded Sets, Center for the Study of Language and Information, Stanford,
    →California, 1988, ISBN 0937073229.
[4] ITO, K. (editor). Encyclopedic Dictionary of Mathematics, Mathematical Society of Japan, Edition 2,
    →Cambridge, Massachusetts, MIT Press, 1993, ISBN 0262590204.",
        lci:remodelledPart2 "ClassOfAbstractObject: A [class_of_abstract_object] is a [class] whose members
            →classify members of [abstract_object].",
        lci:remodelledPart2 "AbstractObject: An [abstract_object] is a [thing] that does not exist in
            →space-time.",
        lci:remodelledPart2 "ClassOfInformationObject: A [class_of_information_object] is a
            →[class_of_arranged_individual] whose members are members of zero or more
            →[class_of_information_representation] and of zero or more
            →[class_of_information_presentation].",
```

```
        lci:remodelledPart2 "ClassOfClass: A [class_of_class] is a [class] whose members are instances of
            →[class].",
        lci:remodelledPart2 "ClassOfClass: NOTE When it is necessary to classify a [class_of_class], another
            →[class_of_class] can be used. This is because a [class_of_class] is a [class].",
        rdfs:label "InformationObject",
        lci:remodelledPart2 "ClassOfRepresentationOfThing: EXAMPLE The [class_of_relationship] that
            →indicates that occurrences of the pattern denoted by 'London' represent the concept of the
            →capital of the United Kingdom can be represented by an instance of
            →[class_of_representation_of_thing].",
        lci:seeAlsoPart2 "DocumentDefinition: A [document_definition] is a
            →[class_of_class_of_information_representation] that defines the content and/or structure of
            →documents.",
        lci:remodelledPart2 "Class: EXAMPLE 1 Centrifugal pump is a [class].
EXAMPLE 2 Mechanical equipment type is a [class].
EXAMPLE 3 Temperature is a [class].
EXAMPLE 4 Commercial fusion reactor is a [class].
EXAMPLE 5 Centigrade scale is a [class].",
        lci:remodelledPart2 "Class: A [class] is a [thing] that is an understanding of the nature of things
            →and that divides things into those which are members of the class and those which are not
            →according to one or more criteria.
The identity of a [class] is ultimately defined by its members. No two classes have the same membership.
    →However, a distinction must be made between a [class] having members, and those members being known,
    →so within an information system the members recorded may change over time, even though the true
    →membership does not change.",
        lci:seeAlsoPart2 "DocumentDefinition: EXAMPLE XYZ Corp. Material Safety Data Sheet is a
            →[document_definition].",
        lci:remodelledPart2 "ClassOfInformationObject: NOTE Usually, it is a physical_object (like a paper
            →document) that is classified as a [class_of_information_object].",
        lci:remodelledPart2 "ClassOfInformationObject: EXAMPLE Newspaper is a
            →[class_of_information_object].",
        lci:remodelledPart2 "ClassOfRepresentationOfThing: A [class_of_representation_of_thing] is a
            →[class_of_relationship] that indicates that all members of the pattern
            →[class_of_information_representation] represent the [thing]."


Class: lci:Organisation

    Annotations:
        lci:equivalentPart2 "ClassOfOrganization: EXAMPLE Company, government, and project team can be
            →represented by instances of [class_of_organization]",
        rdfs:label "Organisation",
        lci:equivalentPart2 "ClassOfOrganization: A [class_of_organization] is a
            →[class_of_arranged_individual] whose members are instances of [physical_object] that are
            →composed of temporal parts of people and other assets, and are organised with a particular
            →purpose."


Class: lci:Organism

    Annotations:
        rdfs:label "Organism",
        lci:equivalentPart2 "ClassOfOrganism: A [class_of_organism] is a [class_of_arranged_individual]
            →whose members are living organisms. ",
        lci:equivalentPart2 "ClassOfOrganism: EXAMPLE Human being, sheep, earthworm, oak tree, and bacteria
            →are instances of [class_of_organism]."

    SubClassOf:
        lci:PhysicalObject


Class: lci:PeriodInTime
```

```
    Annotations:
        lci:equivalentPart2 "PeriodInTime: A [period_in_time] is a [possible_individual] that is all space
            →for part of time - a temporal part of the universe.",
        lci:equivalentPart2 "PeriodInTime: EXAMPLE 1 July 2000 is an instance of [period_in_time].
EXAMPLE 2 The period described by UTC 2000-11-21T06:00 to UTC 2000-11-21T11:53 is an instance of
    →[period_in_time] compliant with ISO8601.",
        lci:equivalentPart2 "ClassOfPeriodInTime: EXAMPLE Monday and June are examples of
            →[class_of_period_in_time].",
        rdfs:label "PeriodInTime",
        lci:equivalentPart2 "ClassOfPeriodInTime: A [class_of_period_in_time] is a [class_of_individual]
            →whose members are instances of [period_in_time]."

    SubClassOf:
        lci:Activity


Class: lci:Person

    Annotations:
        rdfs:label "Person",
        lci:equivalentPart2 "ClassOfPerson: EXAMPLE Engineer, plant manager, student, male, female, senior
            →citizen, adult, girl, and boy can be represented by instances of [class_of_person]. Engineer,
            →plant manager, and student are also instances of [class_of_functional_object].",
        lci:equivalentPart2 "ClassOfPerson: A [class_of_person] is a [class_of_organism] whose members are
            →people."

    SubClassOf:
        lci:Organism


Class: lci:Phase

    Annotations:
        lci:equivalentPart2 "Phase: EXAMPLE The classes known as 'liquid' and 'solid' can be represented by
            →instances of [phase].",
        lci:equivalentPart2 "Phase: NOTE [phase] excludes types of internal structure such as crystalline. ",
        lci:equivalentPart2 "Phase: 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.",
        rdfs:label "Phase"

    SubClassOf:
        lci:InanimatePhysicalObject


Class: lci:PhysicalObject

    Annotations:
        lci:deprecatedPart2 "MaterializedPhysicalObject: A [materialized_physical_object] is a
            →[physical_object] that has matter and/or energy continuity as its basis for identity. Matter
            →or energy continuity requires some matter or energy to be common to adjacent temporal parts of
            →the [materialized_physical_object]. Replacement of some components from time to time does not
            →create a new identity.",
        lci:equivalentPart2 "PhysicalObject: EXAMPLE 1 A piece of metal is a [physical_object].
EXAMPLE 2 A tree is a [physical_object]
EXAMPLE 3 The thing identified by tag P101 is a [physical_object].
EXAMPLE 4 A light beam is a [physical_object].
EXAMPLE 5 A tank that is built and dismantled on site is both a [materialized_physical_object] and a
    →[functional_physical_object].",
        lci:deprecatedPart2 "MaterializedPhysicalObject: EXAMPLE The shell and tube heat exchanger with
            →manufacturer's serial number ES/1234 can be represented by an instance of
            →[materialized_physical_object].",
        rdfs:label "PhysicalObject",
```

```
        lci:equivalentPart2 "PhysicalObject: A [physical_object] is a [possible_individual] that is a
            →distribution of
matter, energy, or both. "


Class: lci:PhysicalQuantity

    Annotations:
        lci:remodelledPart2 "Property: NOTE 1 A member of a [property] is a [possible_individual] that has
            →the same degree or magnitude of the quality or characteristic represented by the [property] as
            →other members.
NOTE 2 The types of characteristic or quality, such as temperature or density, are instances of
    →[class_of_property].
NOTE 3 Duplicate properties (e.g. that map to the same number on the same scale) should not be created
    →within the same data store.",
        rdfs:label "PhysicalQuantity",
        lci:remodelledPart2 "PropertySpace: A [property_space] is a [class_of_property] whose members are a
            →coherent continuum of [property].",
        lci:remodelledPart2 "SinglePropertyDimension: EXAMPLE Temperature, pressure, viscosity, and length
            →are examples of [single_property_dimension].",
        lci:remodelledPart2 "ClassOfPropertySpace: A [class_of_property_space] is a [class_of_class] whose
            →members are members of [property_space].",
        lci:remodelledPart2 "ClassOfProperty: EXAMPLE 'Temperature' is an example of [class_of_property].",
        lci:remodelledPart2 "SinglePropertyDimension: A [single_property_dimension] is a [property_space]
            →that is a single and complete continuum of properties each of which maps to a single number.",
        lci:remodelledPart2 "PropertySpace: EXAMPLE 1 The set of temperature properties, known as
            →temperature, is a [property_space].
EXAMPLE 2 The members of the pressure and flow rate [class_of_property] that fall on a particular pump
    →curve is a [property_space].",
        lci:remodelledPart2 "PropertyRange: EXAMPLE -10C to +20C is a [property_range] of temperature.",
        lci:remodelledPart2 "PropertyRange: A [property_range] is a [property_space] that is a continuous
            →subset of a [single_property_dimension].",
        lci:remodelledPart2 "ClassOfProperty: A [class_of_property] is a [class_of_class_of_individual]
            →whose members are instances of [property]. ",
        lci:remodelledPart2 "ClassOfPropertySpace: EXAMPLE 1 Property curves, property areas, and property
            →volumes of various dimensionality and degrees of freedom are members of
            →[class_of_property_space].
EXAMPLE 2 Pump performance curve is an example of [class_of_property_space].",
        lci:remodelledPart2 "Property: A [property] is a [class_of_individual] that is a member of a
            →continuum of a [class_of_property]. The [property] may be quantified by mapping to a number on
            →a scale.",
        lci:remodelledPart2 "Property: EXAMPLE A particular degree of hotness can be represented as an
            →instance of [property]."

    SubClassOf:
        lci:Quality


Class: lci:PointInSpace

    Annotations:
        rdfs:label "PointInSpace"

    SubClassOf:
        lci:SpatialLocation


Class: lci:PointInTime

    Annotations:
        rdfs:label "PointInTime",
```

```
        lci:equivalentPart2 "PointInTime: EXAMPLE The time known as UTC 1999-05-13T16:31:23.56 is a
            →[point_in_time].",
        lci:equivalentPart2 "ClassOfPointInTime: EXAMPLE Midnight is a [class_of_point_in_time]",
        lci:equivalentPart2 "PointInTime: NOTE  In using this part of ISO15926, a [point_in_time] should be
            →represented by a [representation_of_Gregorian_date_and_UTC_time].",
        lci:equivalentPart2 "ClassOfPointInTime: A [class_of_point_in_time] is a [class_of_event] whose
            →members are members of [point_in_time]. ",
        lci:equivalentPart2 "PointInTime: An [event] that is the whole space extension with zero extent in
            →time."

    SubClassOf:
        lci:Event


Class: lci:Quality

    Annotations:
        rdfs:label "Quality"


Class: lci:QuantityDatum

    Annotations:
        rdfs:label "QuantityDatum",
        rdfs:comment "This class is inspired by the class \"measurement datum\" of the Information Artefact
            →Ontology. The change of wording from \"measurement\" to \"quantity\" is intended to support
            →cases where measurement is not involved, such as with nominal values."

    SubClassOf:
        lci:InformationObject


Class: lci:RegionInSpace

    Annotations:
        rdfs:label "RegionInSpace"

    SubClassOf:
        lci:SpatialLocation


Class: lci:Role

    Annotations:
        lci:seeAlsoPart2 "PossibleRoleAndDomain: EXAMPLE Acting as an anchor is a possible role for pump
            →1234.",
        lci:equivalentPart2 "Role: EXAMPLE 1 Employee is a [role] that indicates what a temporal part of a
            →person has to do with an employment relation.
EXAMPLE 2 Pumper is a [role] that indicates what a temporal part of a pump has to do with a pumping
    →activity.",
        lci:seeAlsoPart2 "ClassOfPossibleRoleAndDomain: EXAMPLE Pumps can play the [role] of anchor
            →(although they are not intended to do so).",
        rdfs:comment "This class is motivated in the Part 2 'role' entity type, and in the same-named BFO
            →class. Part 2 is not very specific about the meaning of roles, but the examples are clear
            →enough. There is still much disagreement in the ontology field about how roles should be
            →understood and modelled.",
        lci:remodelledPart2 "RoleAndDomain: A [role_and_domain] is a [class] that specifies the domain and
            →role for an end of a [class_of_relationship] or [class_of_multidimensional_object].",
        rdfs:label "Role",
        lci:seeAlsoPart2 "IntendedRoleAndDomain: An [intended_role_and_domain] is a [relationship] that
            →indicates the [role_and_domain] some temporal part of the [possible_individual] is intended to
            →take with respect to some [activity].",
```

```
        lci:remodelledPart2 "RoleAndDomain: EXAMPLE \"Husband and man\" and \"wife and woman\" are examples
            →of [role_and_domain].",
        lci:remodelledPart2 "RoleAndDomain: NOTE A [role_and_domain] is analogous to specifying an EXPRESS
            →attribute or its inverse.",
        lci:seeAlsoPart2 "PossibleRoleAndDomain: A [possible_role_and_domain] is a [relationship] that
            →indicates that a player [possible_individual] can possibly play the played [role_and_domain].",
        lci:seeAlsoPart2 "IntendedRoleAndDomain: EXAMPLE Some [possible_individual] that is classified as a
            →pump is intended to play the [role_and_domain] of a performer in some pumping activity.",
        lci:seeAlsoPart2 "ClassOfIntendedRoleAndDomain: A [class_of_intended_role_and_domain] is a
            →[class_of_relationship] that indicates that a member of the [class_of_individual] is intended
            →to act as a member of the [role_and_domain].",
        lci:seeAlsoPart2 "ClassOfIntendedRoleAndDomain: EXAMPLE Pumps are intended to play the
            →[role_and_domain] of performer in some pumping activity.",
        lci:equivalentPart2 "Role: A [role] is a [role_and_domain] that indicates what some thing has to do
            →with an [activity], [relationship], or [multidimensional_object].",
        lci:seeAlsoPart2 "ClassOfPossibleRoleAndDomain: A [class_of_possible_role_and_domain] is a
            →[class_of_relationship] that indicates the [role_and_domain] that can be played by a member of
            →the [class_of_individual], in some [activity]."


Class: lci:ScalarQuantityDatum

    Annotations:
        lci:seeAlsoPart2 "ClassOfIndirectProperty: A [class_of_indirect_property] is a
            →[class_of_relationship] that indicates that a member of the [class_of_individual] can possess
            →a member of the [class_of_property] as an [indirect_property] of this type.",
        lci:seeAlsoPart2 "ComparisonOfProperty: EXAMPLE That the temperature in a room is less than that in
            →a furnace can be indicated by an instance of [comparison_of_property].",
        lci:remodelledPart2 "EnumeratedPropertySet: An [enumerated_property_set] is a [class_of_property]
            →and an [enumerated_set_of_class] whose members are an enumerated set of properties of the same
            →[single_property_dimension] or [multidimensional_property_space].",
        lci:seeAlsoPart2 "IndirectProperty: An [indirect_property] is a [relationship] between a [property]
            →and a [possible_individual]. The nature of the [indirect_property] is defined by its
            →[classification] by a [class_of_indirect_property]. A property is indirect when it does not
            →directly apply to the [possible_individual] it applies to, but is derived from some process.",
        lci:remodelledPart2 "LowerBoundOfPropertyRange: EXAMPLE The instance of [property] that is
            →represented by the instance of [EXPRESS_real] '-10' has a [lower_bound_of_property_range]
            →relationship with the instance of [property_range] '(-10 to +20 degrees Celsius)'.",
        lci:seeAlsoPart2 "ComparisonOfProperty: A [comparison_of_property] is a [relationship] that
            →indicates the magnitude of one [property] is greater than that of another.",
        lci:remodelledPart2 "MultidimensionalProperty: A [multidimensional_property] is a [property] that is
            →also a [multidimensional_object].",
        lci:seeAlsoPart2 "MultidimensionalPropertySpace: EXAMPLE A pump performance curve of flowrate and
            →differential head is a [multidimensional_property_space].",
        lci:remodelledPart2 "UpperBoundOfPropertyRange: An [upper_bound_of_property_range] is a
            →[classification] that indicates that the [property] is the upper bound of the
            →[property_range].",
        lci:remodelledPart2 "LowerBoundOfPropertyRange: A [lower_bound_of_property_range] is a
            →[classification] that indicates that a [property] is the lower bound of a [property_range].",
        lci:remodelledPart2 "MultidimensionalProperty: EXAMPLE A pump flow head characteristic is a
            →[multidimensional_property]. It consists of a continuum of Q, H property pairs, where Q is the
            →flow rate and H is the flowing head difference. Each pair of properties Qa and Ha, where Qa is
            →a particular flow rate and Ha a particular head, is a [multidimensional_property] (Qa, Ha).",
        lci:remodelledPart2 "UpperBoundOfPropertyRange: EXAMPLE +20 Celsius is the upper bound of the range
            →-10 to +20 Celsius.",
        lci:seeAlsoPart2 "Recognition: EXAMPLE Measurement activity #358 recognized that the room was a
            →member of the 20 Celsius [property].",
        lci:seeAlsoPart2 "IndirectProperty: NOTE A property is indirect because it does not directly apply.
            →There can only be one temperature that a thing has (at a time), so a Maximum Allowable Working
            →Temperature is not its temperature, but an indirect property derived from doing some tests or
            →calculations to determine its value (as opposed to it being a current measurement). This is
            →what makes it indirect.",
```

```
        lci:remodelledPart2 "MultidimensionalScale: A [multidimensional_scale] is a [scale] that is also a
            →[multidimensional_object].",
        rdfs:comment "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.",
        lci:remodelledPart2 "MultidimensionalScale: EXAMPLE A [Celsius, seconds] scale is a
            →[multidimensional_scale] on which temperature variation over time can be plotted.",
        lci:seeAlsoPart2 "Recognition: A [recognition] is a [relationship] that indicates that a [thing] is
            →recognized through an [activity].",
        lci:seeAlsoPart2 "IndirectProperty: EXAMPLE A Maximum Allowable Working Pressure of 50 BarA for V101
            →is specified by an [indirect_property] between the pressure of 50 BarA and V101, classified by
            →the [class_of_indirect_property] Maximum Allowable Working Pressure.",
        lci:seeAlsoPart2 "MultidimensionalPropertySpace: A [multidimensional_property_space] is a
            →[property_space] and a [multidimensional_object] whose members are properties each of which
            →maps to more than one number. Each property will consist of elements of the same property
            →dimensions.",
        rdfs:label "ScalarQuantityDatum",
        lci:seeAlsoPart2 "ClassOfRecognition: A [class_of_recognition] is a [class_of_relationship] that
            →indicates that a member of a [class_of_activity] may result in the recognition of a member of
            →a [class].",
        lci:remodelledPart2 "EnumeratedPropertySet: EXAMPLE {115 Volt, 240 Volt} is an example of an
            →[enumerated_property_set].",
        lci:seeAlsoPart2 "ClassOfRecognition: EXAMPLE A measurement activity may result in the recognition
            →of the [classification] of a [possible_individual] by a [property].",
        lci:seeAlsoPart2 "ClassOfIndirectProperty: EXAMPLE Maximum Allowable Working Pressure is a
            →[class_of_indirect_property] that is indicated by a pressure, and can be possessed by a
            →pressure vessel."

    SubClassOf:
        lci:QuantityDatum,
        lci:datumUOM some lci:UnitOfMeasure,
        lci:datumValue some rdfs:Literal


Class: lci:Scale

    Annotations:
        lci:equivalentPart2 "ClassOfScale: EXAMPLE SI Unit is an example of class_of_scale.",
        lci:seeAlsoPart2 "ClassOfScaleConversion: A [class_of_scale_conversion] is a
            →[class_of_isomorphic_functional_mapping] that defines a conversion between two different
            →scales of units used for the quantification of properties.",
        lci:equivalentPart2 "ClassOfScale: A [class_of_scale] is a [class_of_class_of_relationship] whose
            →members are instances of [scale].",
        lci:equivalentPart2 "Scale: EXAMPLE The link that is known as the Celsius scale between the
            →[class_of_number] [-273, inf] and the [class_of_property] temperature can be represented by an
            →instance of [scale].",
        lci:seeAlsoPart2 "ClassOfScaleConversion: EXAMPLE The Fahrenheit scale for temperature and the
            →Celsius scale for temperature can each be represented by instances of [scale]. The conversion
            →between these scales can be represented by an instance of [class_of_scale_conversion].",
        rdfs:label "Scale",
        lci:equivalentPart2 "Scale: A [scale] is a [class_of_isomorphic_functional_mapping] whose members
            →are members of [property_quantification]. It indicates the [number_space] a [property_space]
            →maps to for the [scale] in question."

    SubClassOf:
        lci:UnitOfMeasure


Class: lci:Site

    Annotations:
        rdfs:comment "This class is inspired by the class \"site\" of the Information Artefact Ontology.",
```

```
        rdfs:label "Site",
        rdfs:comment "From BFO: \"b is a site means: b is a three-dimensional immaterial entity that is
            →(partially or wholly) bounded by a material entity or it is a three-dimensional immaterial
            →part thereof. (axiom label in BFO2 Reference: [034-002])\""


Class: lci:SpatialLocation

    Annotations:
        rdfs:label "SpatialLocation",
        lci:equivalentPart2 "SpatialLocation: EXAMPLE Geographic datum, license block, construction area,
            →country, air corridor, maritime traffic zone, hazard control zone, 4D points, lines, planes,
            →solids.",
        lci:equivalentPart2 "SpatialLocation: A [spatial_location] is a [physical_object] that has
            →continuity of relative position."


Class: lci:Stream

    Annotations:
        lci:equivalentPart2 "Stream: EXAMPLE Flux is a 4D-constrained case of [stream] where the path
            →crosses a surface.
EXAMPLE The naphtha flowing in a pipe between a crude distillation unit and a platformer is a [stream].",
        rdfs:label "Stream",
        lci:equivalentPart2 "Stream: 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."

    SubClassOf:
        lci:InanimatePhysicalObject


Class: lci:UnitOfMeasure

    Annotations:
        rdfs:label "UnitOfMeasure"


DisjointClasses:
        lci:Activity,lci:InformationObject,lci:Organisation,lci:PhysicalObject,
            →lci:Quality,lci:Site,lci:SpatialLocation,lci:UnitOfMeasure
```