

2.4. Content for the OWL Axiom Refset

Axioms are statements or propositions which are regarded as being established, accepted, or self-evidently true in the domain. The OWL axioms are in the scope for the OWL axiom refset if they are allowed in the [SNOMED CT Logic Profile Specification](#). Annotation axioms and Class declarations are generally excluded from the OWL axiom refset to avoid duplication to the RF2 files:

- Annotation Axiom - Descriptions can be represented as annotation assertions. They are excluded from the OWL axiom refset to avoid duplication to the RF2 description file.

The transformation from the OWL expression refset to OWL ontology document should process the description file and language refset when descriptions are included. They should be represented by the following annotation properties for a given language refset.

- `rdfs:label` - if present, it is RF2 fully specified name
- `skos:prefLabel` - if present, it is RF2 preferred term
- `skos:altLabel` - if present, it is RF2 acceptable synonym
- `skos:definition` - if present, it is RF2 definition
- Declaration
 - Declarations for the built-in entities, e.g. `owl:Thing`, `owl:Nothing`, are excluded from the OWL refsets because they are implicitly presented in every OWL 2 ontology.
 - Class and property declarations are excluded from the OWL axiom refset to avoid duplication to the content that is represented by the RF2 concept file. Note, Class and property declarations should be included in cases where an entity type cannot be derived from the existing axiom.

Referenced component for an Axiom

The OWL Axiom Reference Set is designed to cover all logic definitions in SNOMED CT. A concept can be defined by one or more axioms in the same module or different modules. Each axiom is represented by a string in [OWL Functional Syntax](#) in the `owlExpression` field. Each concept is represented by a concept ID in the `referencedComponentId` in the refset. The `referencedComponentId` has no impact on the semantics of the axioms and it is not used for any logical reasoning. The following rules should be followed when assigning the `referencedComponentId` for an axiom:

- If an axiom is in the form of `SubClassOf(C D)` or `EquivalentClasses(C D)` where concept C is a precoordinated concept, the concept ID of C should be the `referencedComponentId` for this axiom.
- If an axiom is in the form of `SubPropertyOf(r s)` or `EquivalentProperties(r s)` where attribute r is a precoordinated concept, the concept ID of r should be the `referencedComponentId` for this axiom.
- If an axiom is in the form of `SubClassOf(C D)` where concept C is NOT a precoordinated concept and D is a precoordinated concept. This is a General Concept Inclusion (GCI) axiom. Because concept C is a sufficient but not necessary condition for concept D, the ID of concept D should be the `referencedComponentId` for this axiom. Note, if C is a precoordinated concept, this `SubclassOf` axiom is NOT a GCI axiom and the concept D should not be assigned as the `referencedComponentId`.
- If an axiom is in the form of `SubObjectPropertyOf(u t)` where attribute u is NOT a precoordinated concept and attribute t is a precoordinated concept. The concept ID of t should be the `referencedComponentId` for this axiom. e.g. `SubObjectPropertyOf(ObjectPropertyChain(:r :s) :t)`
- If an axiom is in the form of `SubClassOf(C D)` or `EquivalentClasses(C D)` where both C and D are NOT precoordinated concepts, this is a GCI axiom and the `referencedComponentId` should be 733929006 |General concept inclusion axiom (metadata)|.
- If an axiom is in the form of `DisjointClasses(C D)` where both C and D are precoordinated concepts, the concept ID of C should be the `referencedComponentId` for this axiom. Then the axiom `DisjointClasses(D C)` is redundant.
- The disjointness in more than two Classes can be represented as `DisjointClasses(C D E ...)` where all classes are precoordinated concepts. The concept 787776007 |Disjoint classes axiom (OWL metadata concept)| should be the `referencedComponentId` for this axiom.

The definition status of a concept is |Sufficiently defined by necessary conditions definition status| if the concept has at least one `EquivalentClasses` axiom, irrespective of whether other `SubClassOf` or `EquivalentClasses` axioms exist.

Representation of |is a| Relationships

116680003 |Is a (attribute)| relationships in SNOMED CT are represented by different axioms, `SubClassOf`, `SubObjectProperty`, `SubDataProperty`, or `EquivalentClasses` in the OWL axiom refset.

- `SubClassOf` represents the |Is a (attribute)| relationship between concepts in SNOMED CT.
- `SubObjectPropertyOf` or `SubDataPropertyOf` represents the |Is a (attribute)| relationship between attributes in SNOMED CT for concept model object attributes and concept model data attributes respectively.
- `EquivalentClasses` means that two concepts are subclass of each other in description logics, e.g. `SubClassOf(C D)` and `SubClassOf(D C)`. `EquivalentClasses` are usually used to represent the equivalence between a precoordinated concept (a named class) and an expression such as `ObjectIntersectionOf()` that has one part of a precoordinated superconcept and the other part is an expression that usually refines the superconcept, e.g. `ObjectSomeValuesFrom()` or intersection of expressions.

Class and property are all uniquely identified by an IRI in OWL 2 Ontology. It is not allowed to represent |Is a| relationships between SNOMED CT attributes and concepts in the OWL refset.

- 410662002 |Concept model attribute| should be represented as a class in the OWL axiom refset and SNOMED Ontology.
- 762705008 |Concept model object attribute| and its subconcepts should be represented as an object property in the OWL axiom refset and SNOMED Ontology. The |Is a| relationships among them should be represented as `SubObjectProperty()`;

- 762706009 | Concept model data attribute| and its subconcepts should be represented as a data property in the OWL axiom refset and SNOMED Ontology. The relationship among them should be represented as SubDataProperty();

The metamodeling capacities of OWL 2 allow the punning for Classes and Properties. Both attribute concepts, 762706009 | Concept model data attribute (attribute)| and 762705008 | Concept model object attribute (attribute)|, can be treated as Classes and Properties at the same time to enable the single root node of SNOMED CT. The following two SubClass axioms have been added in addition to SubProperty axioms in the OWL expression refset. These SubClass axioms represent and infer the 116680003 | Is a| relationships to 410662002 | Concept model attribute (attribute)| that is a Class.

- SubClassOf(:762706009 :410662002)
- SubClassOf(:762705008 :410662002)

For inferred relationships in the Necessary Normal Form (NNF), SubClassOf, SubObjectProperty and SubDataProperty in the OWL axiom refset should be represented as | Is a| relationships. The explanation for the Necessary Normal Form and the rules for calculating the NNF can be found in section 2.5. [Generating Necessary Normal Form Relationships from the OWL Refsets.](#)

Representation of concrete domains

DataHasValue class expression should be used for a 762706009 | Concept model data attribute (attribute)| with a particular literal value that is specified in the [SNOMED OWL Logic profile specification](#). The DataSomeValuesFrom and DataAllValuesFrom class expressions should not be used. For example, the value of 1142135004 | Has presentation strength numerator value (attribute)| should be represented as:

```
DataHasValue(:1142135004 "50"^^xsd:decimal)
```

Unfortunately, the xsd:boolean data type is out of the scope for the OWL 2 EL profile (<https://www.w3.org/TR/owl2-profiles/#Entities>). In order to have a consistent representation and clear semantics, the boolean values should be represented by the concept 31874001 | True (qualifier value)| and 64100000 | False (qualifier value)|. The attribute should be a subtype of 762705008 | Concept model object attribute (attribute)| with a constraint in the SNOMED CT Machine Readable Concept Model that only these two values are in the range and the attribute cardinality is 0..1.

Note, trailing zeros are optional for the decimal data type in SNOMED CT. If the fractional part is zero, the period and following zero(es) can be omitted. For example, 2 is equivalent to 2.0. In particular, trailing zeros are prohibited for medicinal products in SNOMED CT because of clinical safety concerns. This 'trailing zero' policy should be applied in the same subject area of SNOMED CT for consistent classifications because the ELK reasoner has an incomplete implementation of concrete domains. However, it is possible that other subject areas, e.g. clinical findings, could allow for trailing zeros if it is desirable.

Attributes

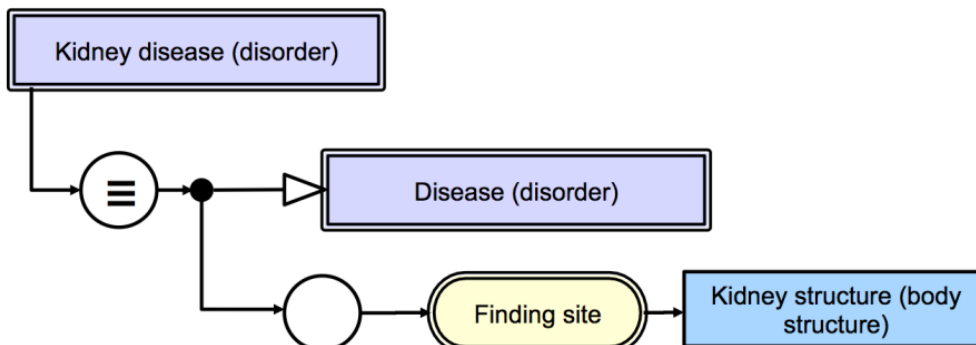
SNOMED CT is based on concepts, with attributes also being represented as concepts. However, attributes need to be represented as properties in the OWL axiom refset. The following rules should be followed.

- The attribute 410662002 | Concept model attribute| should be represented as a Class in the OWL axiom refset.
- The attribute 762705008 | Concept model object attribute (attribute)| and 762706009 | Concept model data attribute (attribute)| should be represented as Classes and Properties as noted above.
- All descendants of 762705008 | Concept model object attribute (attribute)| should be represented as ObjectProperties.
- All descendants of 762706009 | Concept model data attribute (attribute)| should be represented as DataProperties.
- All the rest attributes in SNOMED CT should be represented as Classes.

Role Group

Role groups must be explicitly stated and represented by the concept 609096000 | Role group (attribute)| as an object property in the OWL axiom refset. In the diagram of stated relationships in the OWL axiom refset, the attribute 609096000 | Role group (attribute)| should be represented by a circle rather than an object property. The role group should not be omitted for self-group attributes where there is only a single attribute in a role group.

An example for diagram representation of an OWL axiom



Modules and Axioms

The editing of entries in the OWL refsets can only be performed by the owner of that module. The extensions must not modify the OWL refset entries from the dependent modules, such as [International Health Terminology Standards Development Organization maintained module](#). No changes are permitted to the content of the International Edition, except for the addition of new versions of this content in a module owned by the extension producer.

- **Axiom addition**

SNOMED CT extensions can add new axioms to the concepts in the international release to support extension content, such as adding an extension concept as an additional parent to a concept in the international release. Each axiom in an extension must have a new UUID, moduleId and effectiveTime from the extension. An international concept may have multiple axioms - one or more axiom from the international edition and zero or more axiom from extension. An axiom addition in an extension can be presented by two alternative forms with the same classification result.

- **Axiom inclusion** - a new axiom **with inclusion** of the axiom from the international release as part of the extension axiom. For example, an extension concept D can be added as a superconcept by adding a new axiom in the extension.
SubClassOf(A C) is an axiom released in the international release.
SubClassOf(A ObjectIntersectionOf(C D)) is a new axiom in the extension.
- **Axiom without inclusion** - a new axiom **without inclusion** of the axiom from the international release as part of the extension axiom. For example,
SubClassOf(A C) is an axiom released in the international release.
SubClassOf(A D) is a new axiom in the extension.

It is possible that substantive improvements or corrections to the International Edition can be made through axiom overriding in an extension, if they cannot be achieved by the Axiom addition approach.

- **Axiom overriding** - a new axiom has the same UUID from the international release, but a new effectiveTime, module id and OWL expression from the extension.

This approach is different to the Axiom addition. The extension takes over the ownership of an axiom from the International Edition and overrides the axiom.

Whichever approach is taken by extensions, either Axiom addition or Axiom overriding, it must follow the [General Authoring Principles for extensions](#). Any modifications resulting in changes to the classification of international content must be accompanied by a disclaimer notifying users of the differences between the extension edition and the International Edition. These changes should be forwarded to SNOMED International in a timely fashion to improve the quality of the International Edition for all users. Please note that modifications of this kind pose a risk to the comparability and interoperability of data captured using different SNOMED CT editions.

Versioning for Axioms

The versioning is at the axiom level for a concept in the OWL axiom refset. This means that there is only one effectiveTime for each version of an axiom, which may have multiple relationships. The changes to any relationships in the current editing tool will trigger an "update" of the most recent version of the relevant axiom. The versioning at the axiom level, where each axiom may represent multiple relationships, is different from the versioning of each individual relationship in the relationship file.

Any changes to the NNF should have a corresponding history of changes in the OWL axiom refset. This will ensure that all entries in the NNF can be derived from the OWL refset except for those relationships listed in table 2.4-1. The NNF will still have the computed effectiveTime for each inferred relationship. The version of each inferred relationship can be derived from the OWL refset, but it is not true in reverse.

It is permitted to make modifications to a published axiom without inactivation by the owner of the module, by creating a new version of the axiom with the same UUID and a new effectiveTime. Since any modification to an axiom could potentially alter its meaning, it is not necessary to inactivate an axiom and create a new axiom. It is also not necessary to reinstate an inactivated axiom in the previous release when the same axiom is created as a new expression. This approach can simplify the tooling and authoring process and it is a different approach to the history of changes to the individual relationship.

However, an axiom must be inactivated in the following situations:

1. The concept used as the referenced component is inactivated or changed.
2. Axiom needs to be inactivated without any replacement.
3. Any inactive concept or attribute referenced in the axiom will not be replaced by an active component.

Special notes on axioms during the transitional period from the stated relationship file to the OWL axiom refset

During the transitional period, there should be only one active axiom in the OWL axiom reference set for each concept. This axiom will represent the set of all active attributes (from the existing stated relationship file) for which the given concept is the source concept. This axiom overriding approach should only be used for the transition once. After the transition period, these axioms will be reviewed, and where appropriate will be split into multiple axioms as described in the Axiom addition approach.

If a dependent module (e.g. an extension module) adds defining relationships to a concept, then this will result in a new version of the axiom (which has the new relationship included) being added to the OWL axiom reference set. A new version of the axiom must be created if any of the concept's defining relationships change, regardless of whether or not that change is in the given module. For example (moduleId, sourceId, destinationId, typeId and referencedComponentId are represented by letters for easier readability):

id	effectiveTime	active	moduleId	sourceId	destinationId	typeId
111111	20190731	1	A	X	W	Ra
222222	20190731	1	A	X	Y	Rb
333333	20191031	1	B	X	Z	Rz

Results in two versions of a single OWL reference set entry

UUID	effectiveTime	active	moduleId	referencedComponentId	owlExpression
9c1951e8-bbaa-434b-b9d7-82b460e221de	20190731	1	A	X	EquivalentClasses(:X ObjectIntersectionOf (ObjectSomeValuesFrom(:Ra :W) ObjectSomeValuesFrom(:Rb :Y)))
9c1951e8-bbaa-434b-b9d7-82b460e221de	20191031	1	B	X	EquivalentClasses(:X ObjectIntersectionOf (ObjectSomeValuesFrom(:Ra :W) ObjectSomeValuesFrom(:Rb :Y) ObjectSomeValuesFrom(:Rz :Z)))