Leading healthcare terminology, worldwide

SNOMED CT OWL Guide

Latest web browsable version: http://snomed.org/owl

SNOMED CT document library: http://snomed.org/doc

Publication date: 2018-07-31

Table of Contents

This document is a guide to the use of the Ontology Web Language (OWL) in SNOMED CT. It includes detailed information about structure, content and use of OWL reference sets. These OWL reference sets are are used to distribute ontology reference information and the axioms that represent the formal logical definitions of SNOMED CT concepts.

Web browsable version: http://snomed.org/owl

SNOMED CT Document Library: http://snomed.org/doc

1. Introduction

Background

SNOMED CT is a clinical terminology with a comprehensive coverage of wide clinical specialties and requirements. The development and maintenance of SNOMED CT relies on Description Logics (DL) and its reasoning services. The stated relationship file has been used to represent DL definitions. However, it is unable to fully represent semantics of the DL expressivity due to the limitation of its relational structure.

The new OWL refsets are designed to replace the stated relationship file and it represents the DL definitions for SNOMED CT content by following the international standard of OWL 2 Web Ontology Language.

Purpose

The purpose of this document is to define and describe the OWL Reference Set.

Scope

The document presents the specification of OWL refsets. It also documents the background, design considerations, quality assurance, and distribution format of the OWL refsets.

The implementation of the OWL refsets in the applications, and transformation of it to an OWL ontology, are out of the scope.

Audience

The target audiences of this document include:

- SNOMED CT national release centers;
- SNOMED CT terminology content developers and technical developers;
- SNOMED CT implementers, analysts, and developers of electronic health records and information systems.

2. Design Considerations for OWL Reference Sets

2.1. Reference Set Type

The OWL axioms can be represented following the pattern for annotation type reference set. The Annotation Reference Set allows |String| data type to be associated with components for any specified purpose. However, this approach is not specific enough because OWL expressions would be mixed with other types of annotations. Therefore, a new specific reference set type has been defined.

The |OWL expression type reference set| should only represent components and their associations to OWL expressions, e.g. definitions specified in OWL expressions for concepts, or information about OWL ontologies of SNOMED CT. This will enable accurate representation for feeding data into DL reasoners, validation of OWL expressions and creation of OWL ontology.

2.2. Syntax for the OWL Refsets

The OWL Functional Syntax-Style is the recommended default syntax. The generated OWL ontologies of SNOMED CT may be rendered in any compatible OWL syntax and transformed between syntaxes. The specification of syntax including the BNF grammar published by W3C can be found at https://www.w3.org/TR/owl-syntax/.

Specializations for SNOMED CT

For the benefits of consistent representation and detection of changes to an expression, the style and rules are recommended though they have no semantic significance. The OWL expressions should be rendered in the form specified by the OWL Functional Syntax-Style but a restricted version for creating and updating the OWL refsets should follow the rules listed below.

Whitespace

OWL Functional Syntax-Style defines that whitespace is a nonempty sequence of space (U+20), horizontal tab (U+9), line feed (U+A), or carriage return (U+D) characters.
The whitespace should only be represented as a single nonempty sequence of space (U+20) in the OWL refsets.

Comments

OWL Functional Syntax-Style defines that a comment is a sequence of characters that starts with the # (U+23) character and does not contain the line feed (U+A) or carriage return (U+D) characters.
Comments could improve the readability. However, comments should not be included to minimize the size of the OWL refsets.

Sorting Order

A standard sort order is very useful to enable fast matching of identical expressions though it has no impact to semantics and is not essential for general purpose use.
Each axiom is a serialization of a tree structure that can be traversed by nodes in the following order:

1. Nodes that are a SNOMED CT concept ID, or a value of data property. These nodes are sorted in UTF-8 byte-order (as a sequence of bytes) rather than numbers or strings. The byte-order compares two bytes; one from each sequence, until the values of the bytes are different. Then, the ID or value with the lowest byte value would be sorted as first regardless the length difference of byte arrays;
2. The composite node ObjectSomeValuesFrom(), sorting order by concept id of attribute and then value;
3. The composite node DataHasValue(), sorting order by concept id of attribute and then value.

The Functional-Style Syntax determines the general order of most elements within an OWL axiom. In fact, the order can only be applied to the nodes within EquivalentClasses(), DisjointClasses(), EquivalentDataProperties(), and ObjectIntersectionsOf() for the current SNOMED CT Logic Profile Specification.

2.3. Content for the OWL Ontology Refset

The OWL ontology refset should represent essential information about an ontology, such as the namespaces, ontology URI, ontology version URI, and import statement.
Apart from the standard information for all OWL ontologies, any specific information of an ontology can be included. The OWL ontology refset enables the use of prefixes in the OWL axiom refset.

Namespaces

The namespace declarations cover the standard and SNOMED CT specific prefix names, for all ontologies. The prefix name "sct:" is for SNOMED CT concept identifiers and the namespace URI is http://snomed.info/id/.

The prefix names are associated with SNOMED CT concept 734146004 |OWL ontology namespace (metadata)| as referencedComponentId and examples can be found in table 4-2 OWL Ontology Reference Set example.
The URIs can be represented in full form or using the prefix names. For example, The URI for 64572001 |Disease (disorder)| can be one of the following format.

- http://snomed.info/id/64572001
- :64572001

It is recommended to use the default prefix ":" to minimize the size and improve readability in the OWL refset. The prefix name "sct:" should be declared and used when http://snomed.info/id/ is not the default namespace.

Ontology URI and Version URI

The ontology URI and the version URI together identify a particular version of ontology. According to the convention, an ontology document should be accessible via the ontology URI if it is the current version. Since a release edition contains the current version ontology, the ontology URI alone is sufficient for the OWL ontology refset. The version should be determined from the SNOMED CT release package. The version URI does not need to be included in the OWL ontology refset. For example:

Ontology(<http://snomed.info/sct/900000000000207008>)

SNOMED CT International Edition	http://snomed.info/sct/900000000000207008

However, the version URI should always be included for a standalone ontology file of SNOMED CT to provide accurate version information. The SNOMED CT URI Standard describes how to unambiguously reference a particular version of a SNOMED CT edition. It is also a trivial task to generate the Ontology version URI by the transformation process for a standalone ontology file. For example,

Ontology(<http://snomed.info/sct/900000000000207008> <http://snomed.info/sct/900000000000207008/version/20180731>)

Edition, Modules and Ontology Import Statement

Each SNOMED CT Edition should be represented as a separate ontology based on the identifier of the most dependent module. The international release is represented by a single ontology and identified by the SNOMED CT core module id as the ontology URI. It includes two modules for terminology content excluding modules for derivatives:

- 900000000000207008 |SNOMED CT core module (core metadata concept)|
- 900000000000012004 |SNOMED CT model component module (core metadata concept)|

The module dependency has specified that 900000000000207008 |SNOMED CT core module (core metadata concept)| depends on 900000000000012004 |SNOMED CT model component module (core metadata concept)|.

The OWL ontology reference set should only have one single active entry of |OWL ontology header| for ontology declaration. A new entry for OWL ontology header should be added for an extension edition with the extension module id and a new effective time. The OWL ontology header for SNOMED CT international release should be

inactivated. The identifier for the ontology URI should be the most dependent module for that particular edition. Please refer to the general authoring principles for changes that can be made by extensions.

It is possible that modules in SNOMED CT can be directly translated into OWL ontologies, with the module dependency reference set describing how these ontologies are imported. Extensions can import the ontology of SNOMED CT core module since the content in the model component module has already been included. The new OWL ontology header should include extension module identifier and import statement(s). This approach could avoid accidental changes to the imported ontology. However, the implementation could be more complicated than the representation of an entire edition as a single ontology. Therefore, it is not recommended to use an ontology import statement at the current stage. Instead, each edition will be rendered as one OWL ontology, without any ontology import statement.

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 refset to avoid duplication to the RF2 description file.

The transformation from the OWL 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 refset because they are implicitly presented in every OWL 2 ontology.
    - Class declarations are excluded from the OWL refset to avoid duplication to the content that are represented by the RF2 concept file. Note, Class declaration should be included in cases where an entity type cannot be derived from the existing axiom.

However, property declarations should be included in the OWL refset to reduce the implementation burden for deriving such information at runtime. The property declarations include all attributes for SNOMED CT concept modeling. These attribute concepts must be excluded from the Class declarations.

Concept Defined by 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 association between axiom and concept should follow the following rules:

- 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 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 concept ID of D should be the referencedComponentId for this axiom.

- 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.

The definition status of a concept is |Sufficiently defined concept definition status| if the concept has one EquivalentClasses axiom no matter other associated axioms are SubClassOf or EquivalentClasses.

Representation of |is a| Relationships

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

- SubClassOf represents the |Is a (attribute)| relationship between concepts in SNOMED CT.
- SubObjectProperty or SubDataProperty represents the |Is a (attribute)| relationship between attributes in SNOMED CT.
- 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 precoorrdinated 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 refset and Ontology.
- 762705008|Concept model object attribute (attribute)| and its subconcepts should be represented as object property in the OWL refset and Ontology. The |Is a| relationships among them should be represented as SubObjectProperty();
- 762706009|Concept model data attribute (attribute)| and its subconcepts should be represented as data property in the OWL refset and Ontology. The relationship among them should be represented as SubDataProperty();

For inferred relationships in the Necessary Normal Form (NNF), SubClassOf, SubObjectProperty and SubDataProperty in the OWL 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.
The following additional two |Is a| relationships should always be included in the inferred relationship file because they cannot be derived from the OWL axiom refset.

Table 2.4-1: Additional inferred |Is a| relationships in the NNF file

SourceId	TypeId	DestinationId
762705008 |Concept model object attribute (attribute)|	116680003 |Is a (attribute)|	410662002 |Concept model attribute (attribute)|
762706009 |Concept model data attribute (attribute)|	116680003 |Is a (attribute)|	410662002 |Concept model attribute (attribute)|

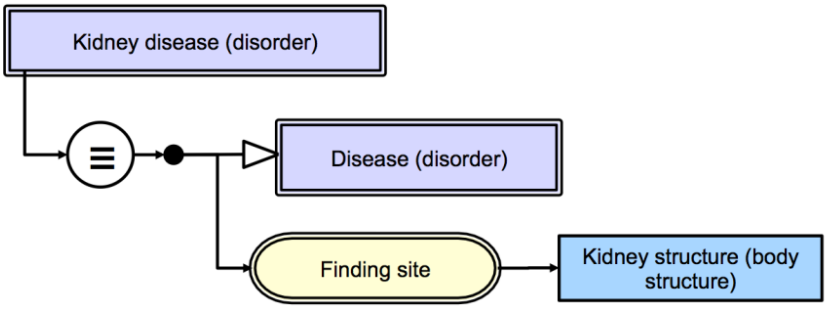
Attribute

SNOMED CT is based on concept and attribute is also concept. However, a property cannot be a subproperty of a class in OWL because class and property are disjoint.
Descendants of 410662002 |Concept model attribute (attribute)| should be represented as either ObjectProperties or DataProperties in the OWL axiom refset. The rest SNOMED CT attributes including 410662002 |Concept model attribute (attribute)| should be represented as Classes in the OWL axiom refset.

Role Group

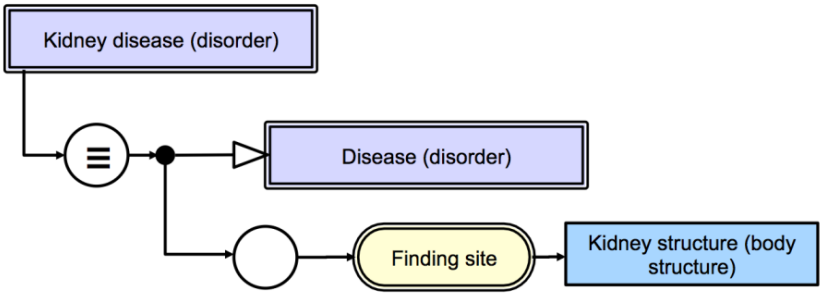
Role group must be explicitly stated and represented by the attribute 609096000 |Role group (attribute)| in the OWL refset. Furthermore, role group should allow having only one attribute and value as "self-grouped". These are key differences to the numeric number representation of role group in the RF2 relationship file.
For diagram of stated relationships in the OWL axiom refset, the attribute 609096000 |Role group (attribute)| should still be represented by a circle rather than an object property.

Current diagram representation for attribute in role group 0



For diagram of relationships in logical model, the circle should be used for 'self-grouped' attribute(s) in the role group 0.

Diagram representation of logical model



Modules and Axioms

The editing of entries in the OWL refsets can only be performed by the owner of that module. In general, the extensions must not modify the OWL refset entries from the dependent modules, such as 900000000000445007| International Health Terminology Standards Development Organization maintained module (core metadata concept)|.
SNOMED CT extensions can add new axioms for the concepts in the international release by following the General Authoring Principles for extensions. A new axiom in the extension module can be represented by two approaches with different semantics.

- **Axiom overriding** - a new axiom **with inclusion** of the axiom from the international release as part of the axiom. The new axiom has the same UUID from the international release with a new effectiveTime and module id in the extension. This approach should be avoided because it overrides the axiom from the dependent module.
- **Multiple axioms** - a new axiom **without inclusion** of the axiom from the international release as part of the axiom. The new axiom has a new UUID and effectiveTime in the extension. This approach provides multiple axioms for an concept in the extension module.

No matter which approach is taken, the transitive closure of the inferred |Is a| relationships should be a superset of transitive closure of the dependent modules in order to ensure 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 refset. It means that there is only one effectiveTime for an axiom that might have multiple relationships. The changes to any relationships in the current editing tool will trigger an "update" of axiom with the latest effectiveTime. They are different to the versioning at individual relationship level in relationship files.

Any changes to the NNF should have a history of changes in the OWL refset. This will ensure that all entries in the NNF can be derived from the OWL refset except relationships in the 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 allowed to make direct modification to a published axiom without inactivation by the owner of the module. The same UUID should be used with 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 individual relationship.

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

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

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

There should be only one active axiom in the reference set for all active relationships in conjunction for a concept at that snapshot across modules, during the period of transformation from the existing stated relationships file to the OWL axiom reference set. Therefore, this is the axiom overriding approach and should only be used for the transformation once. Then, these axioms should be reviewed and many of them could be more suitable for multiple axioms.

If a dependent module (e.g. an extension) adds relationships to that concept, this will result in a new version of the axiom is added to the OWL reference set with the addition. This new version must be created if any of the relationships from the viewpoint of the module's dependencies changes, regardless of whether that change is on the module in question or not. For example,

ID	Module	Version	Role	Source	Target
111	A	1	Ra	X	W
222	A	1	Rb	X	Y
333	B	2	Rz	X	Z

Results in two *versions* of a single OWL reference set entry

UUID	Version	Module	Axiom
1	1	A	EquivalentClasses(:X ObjectIntersectionOf(ObjectSomeValuesFrom(:Ra :W) ObjectSomeValuesFrom(:Rb :Y)))

1	2	B	EquivalentClasses(:X ObjectIntersectionOf(ObjectSomeValuesFrom(:Ra :W) ObjectSomeValuesFrom(:Rb :Y) ObjectSomeValuesFrom(:Rz :Z)))

2.5. Generating Necessary Normal Form Relationships from the OWL Refsets

The logic definitions are represented by the OWL axiom refset that is a replacement of the RF2 stated relationship file. As a result, the nature of the inferred relationship file in the distribution normal form (DNF) has changed, because the new DL features are not representable in the current relationships file. The inferred relationship file will maintain the same format and structure, but it is no longer equivalent to the stated form (containing all necessary and sufficient conditions). In fact, it is a collection of all the necessary conditions of precoordinated concepts and represents a subset of the full semantics.

Necessary Normal Form

The Necessary Normal Form (NNF) is a replacement for the Distribution Normal Form for inferred relationships. The NNF is a precalculated distribution form for practical purposes, for example, to support the continuity of existing implementations based on relational databases and queries by the expression constraint language.

The NNF consists of the full set of necessary relationships of precoordinated concepts after removal of redundant relationships within a given concept definition. Within the scope of a SNOMED CT terminology, necessary relationships are defined only for precoordinated concepts (aka OWL's named classes). Let C be a precoordinated concept and D be either a precoordinated concept or a complex expression. If an axiom is in the form of SubClassOf(C D) or EquivalentClasses(C D), then all of the derivable and necessary relationships of D are necessary relationships of C.

The NNF does not include class disjointness, transitive properties, reflexive properties and sufficient conditions represented as General Concept Inclusions (GCIs) in the OWL axiom refset.

Rules for Determining Redundant Relationships

Rule 1 - Class and Role inclusions

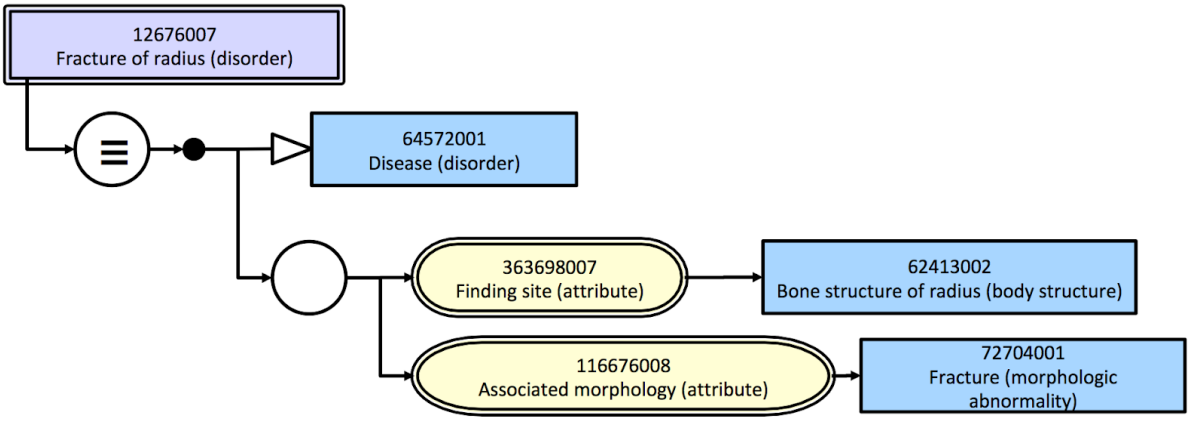
Given two relationships, A and B, A with r = C and B with s = D, within the same role group, A is redundant if:

- r is the same as or a supertype of s, and
- C is the same as or a supertype of D

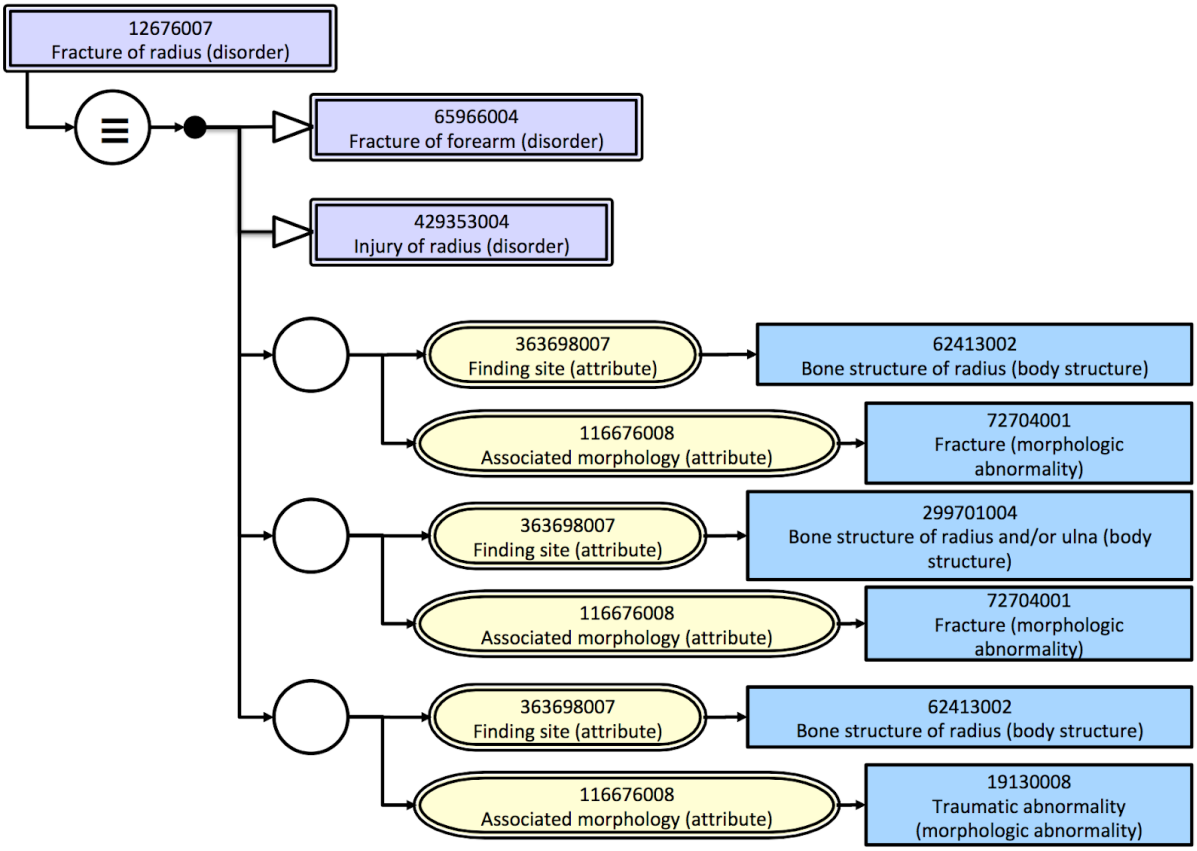
Note, "crossover relationships", where r is a supertype of s, and C is instead a subtype of D do not result in a redundant relationship.

Example for Class inclusion

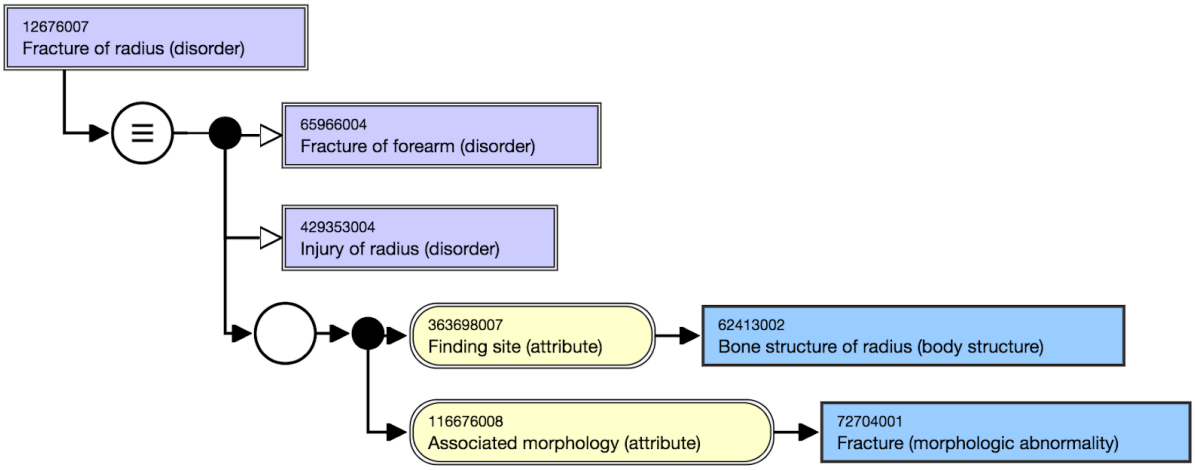
Stated relationships



Inferred relationships before the removal of redundant relationship



Inferred relationships after the reduction



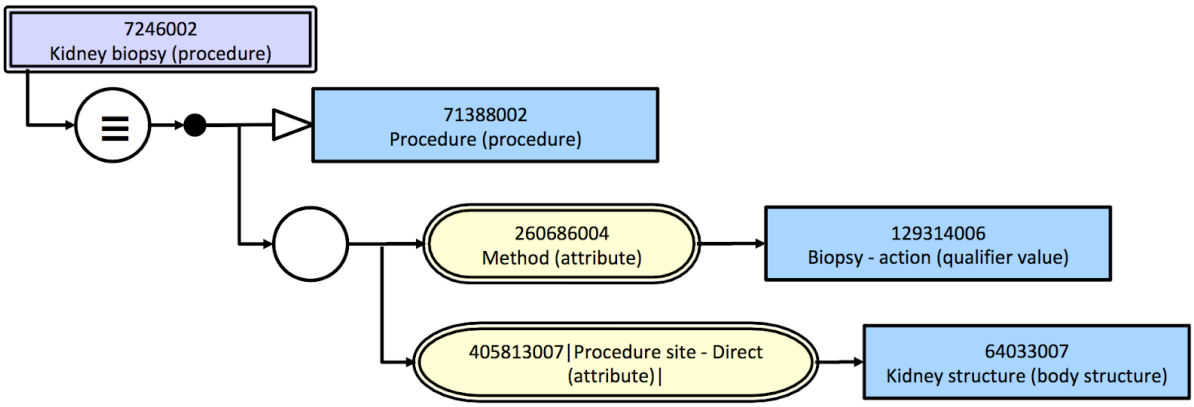
For |Fracture of radius|, the relationship |Finding site| = |Bone structure of radius and/or ulna| is inherited from |Fracture of forearm|, which is a redundant relationship because |Bone structure of radius| is a subtype of |Bone structure of radius and/or ulna|. The relationship |Associated morphology| = |Traumatic abnormality| is inherited from |Injury of radius|, which is a redundant relationship because |Fracture (morphologic abnormality)| is a subtype of |Traumatic abnormality|.

Table: Example in OWL axiom refset and RF2 relationship file (NNF)

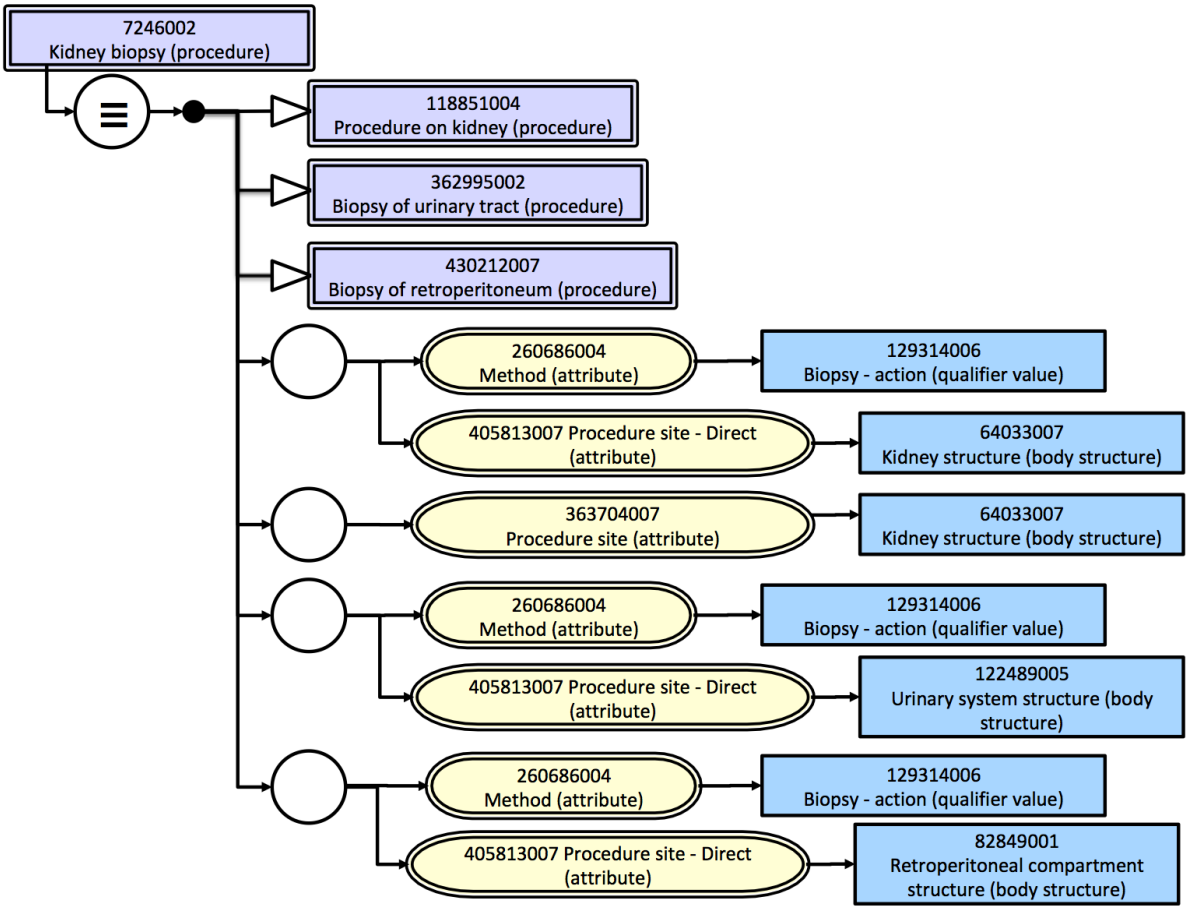
referencedComponentId	owlExpression (stated relationships)	Inferred Relationships in Necessary Normal Form			
		sourceId	destinationId	relationshipGroup	typeId
125605004	EquivalentClasses(:125605004 ObjectIntersectionOf(:64572001 ObjectSomeValuesFrom(:609096000 ObjectIntersectionOf(ObjectSomeValuesFrom(:116676008 :72704001) ObjectSomeValuesFrom(:363698007 :272673000)))))	125605004	284003005	0	116680003
		125605004	72704001	1	116676008
		125605004	272673000	1	363698007
12676007	EquivalentClasses(:12676007 ObjectIntersectionOf(:64572001 ObjectSomeValuesFrom(:609096000 ObjectIntersectionOf(ObjectSomeValuesFrom(:116676008 :72704001) ObjectSomeValuesFrom(:363698007 :62413002)))))	12676007	65966004	0	116680003
		12676007	429353004	0	116680003
		12676007	72704001	1	116676008
		12676007	62413002	1	363698007
62413002	SubClassOf(:62413002 :299701004)	62413002	299701004	0	116680003

Example for Role inclusion

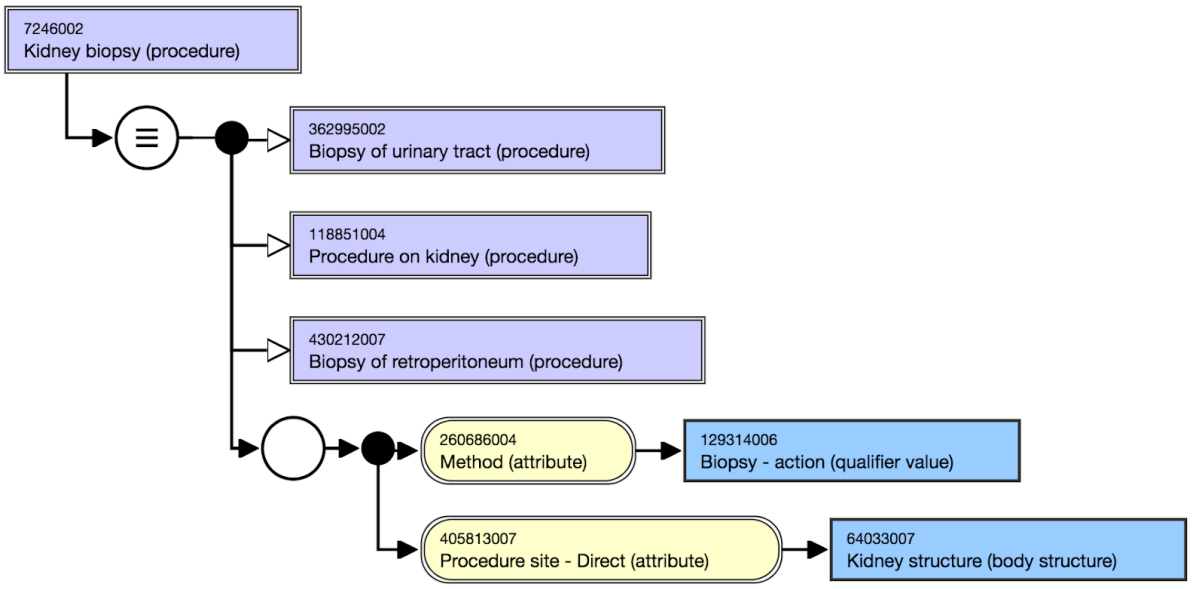
Stated relationships



Inferred relationships before the removal of redundant relationship



Inferred relationships after reduction



For concept |Kidney biopsy|, the relationship |Procedure site| = |Kidney structure| is inherited from |Procedure on kidney|, which is a redundant relationship to |Procedure site - Direct| = |Kidney structure| because |Procedure site - Direct| is a subtype of |Procedure site|. Because |Kidney structure| is a subtype of |Urinary system structure| and |Retroperitoneal compartment structure|, the inherited relationships for |Procedure site - Direct| are also redundant.

Table: Example in OWL axiom refset and RF2 relationship file (NNF)

referencedComponentId	owlExpression (stated relationships)	Inferred Relationships in Necessary Normal Form			
		sourceId	destinationId	relationshipGroup	typeId
118851004	EquivalentClasses(:118851004 ObjectIntersectionOf(:71388002 ObjectSomeValuesFrom(:609096000 ObjectSomeValuesFrom(:363704007 :64033007))))	118851004	713880002	0	116680003
		118851004	64033007	1	363704007
7246002	EquivalentClasses(:7246002 ObjectIntersectionOf(:71388002 ObjectSomeValuesFrom(:609096000 ObjectIntersectionOf(ObjectSomeValuesFrom(:260686004 :129314006) ObjectSomeValuesFrom(:405813007 :64033007)))))	724602	118851004	0	116680003
		724602	362995002	0	116680003
		724602	430212007	0	116680003
		724602	129314006	1	260686004
		724602	64033007	1	405813007
405813007	SubObjectPropertyOf(:405813007 :363704007)	405813007	363704007	0	116680003

Rule 2 - Property chains including transitive properties

Given attribute r, s and t with a property chain SubObjectPropertyOf(ObjectPropertyChain(t s) r), and two relationships A and B, A with r = C and B with u = D, within the same role group, A is redundant if:

- Attribute u is the same as or a subtype of t, and
- D has relationship to C via attribute s

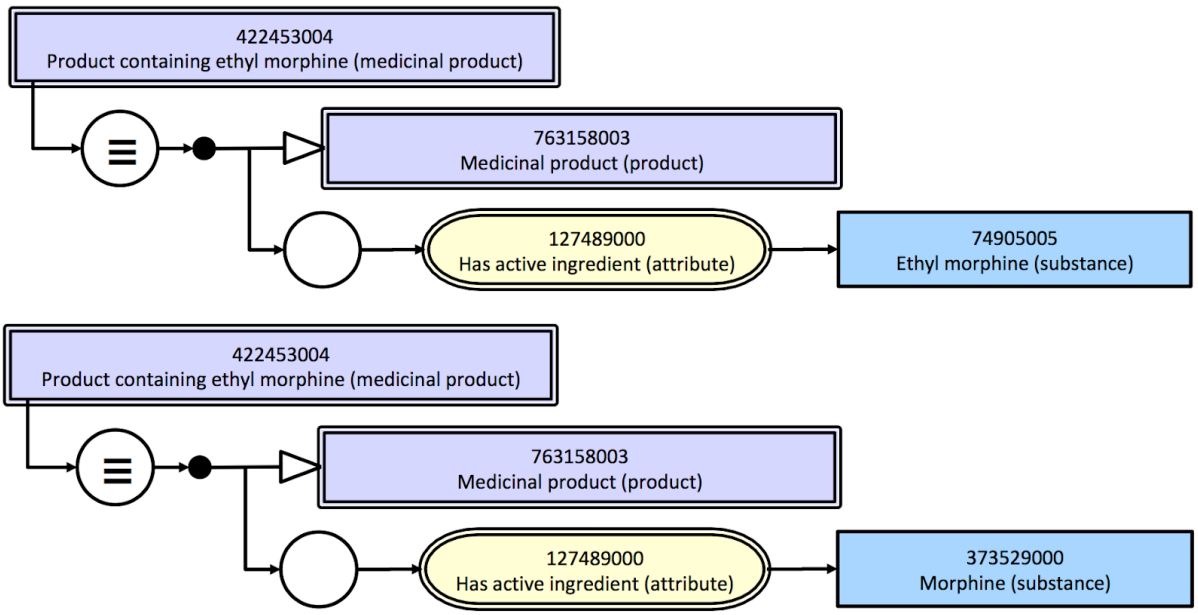
Note the following:

- C does not need to subsume D
- Attribute t does not need to be the same as or a subtype of r
- Transitive properties are defined by a property chain in the form of

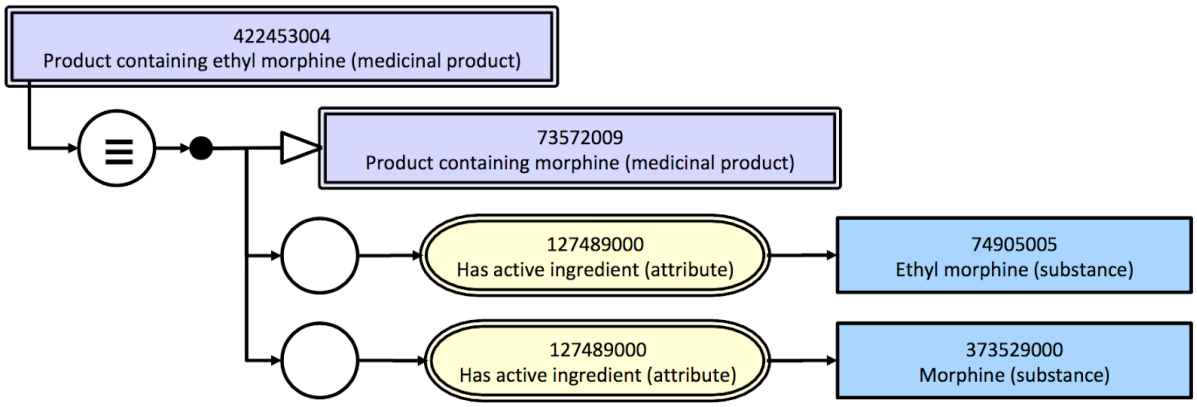
SubObjectPropertyOf(ObjectPropertyChain(r r) r) and thus it is a special case of the above.

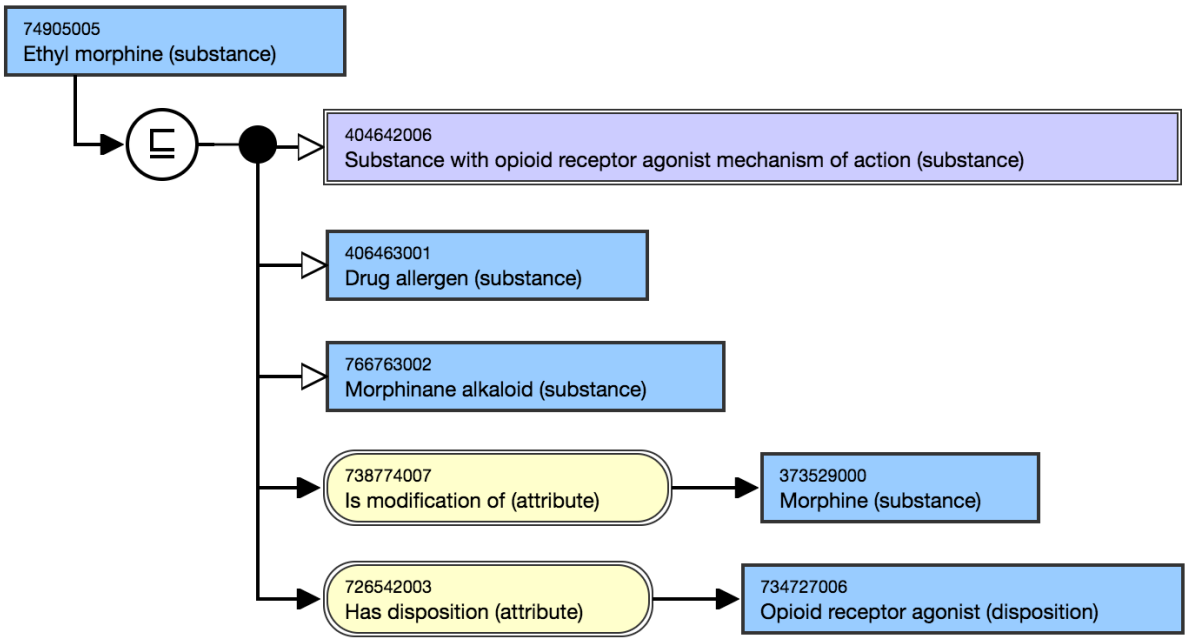
Example for property chain:

Stated relationships

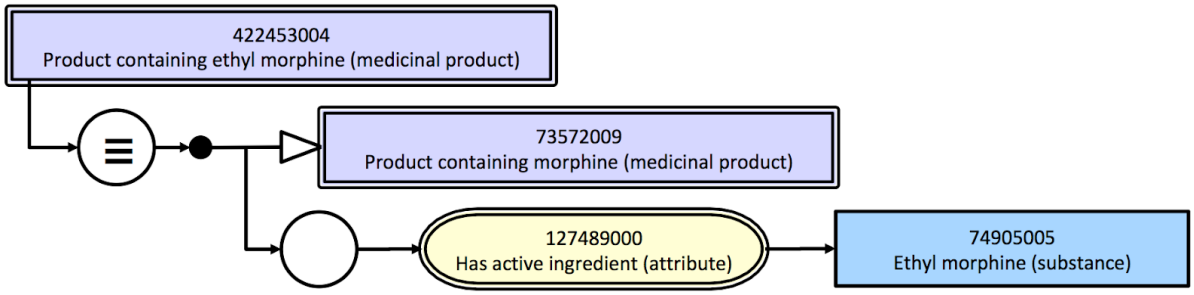


Inferred relationships before the removal of redundant relationship





Inferred relationships after reduction



For |Product containing ethyl morphine|, the relationship |Has active ingredient| = |Morphine| is inherited from |Product containing morphine|, which is a redundant relationship to |Has active ingredient| = |Ethyl morphine| because |Ethyl morphine| |Is modification of| |Morphine| and property chain of |Has active ingredient| and |Is modification of| is a sub-property of |Has active ingredient|.

Table: Example in OWL axiom refset and RF2 relationship file (NNF)

referencedComponentId	owlExpression (stated relationships)	Inferred Relationships in Necessary Normal Form			
		sourceId	destinationId	relationshipGroup	typeId
73572009	EquivalentClasses(:73572009 ObjectIntersectionOf(:763158003 ObjectSomeValuesFrom(:609096000 ObjectSomeValuesFrom(:127489000 :373529000))))	73572009	764887005	0	116680003
		73572009	360204007	0	116680003
		73572009	373529000	1	127489000

422453004	EquivalentClasses(:422453004 ObjectIntersectionOf(:763158003 ObjectSomeValuesFrom(:609096000 ObjectSomeValuesFrom(:127489000 : 74905005))))	422453004	73572009	0	116680003
		422453004	74905005	1	127489000
127489000	SubObjectPropertyOf(ObjectPropertyChain(:127489000 :738774007) : 127489000))	N/A	N/A		N/A
74905005	SubClassOf(:74905005 ObjectIntersectionOf(:440327007 ObjectSomeValuesFrom(:738774007 :373529000)))	74905005	440327007	0	116680003
		74905005	373529000	0	738774007

Technical Implementation for Calculating the NNF

This fairly complex process uses the stated form and the output of the reasoner to calculate the necessary normal form which is represented in the relationship RF2 file.

The most straightforward way to produce the necessary normal form would be to use the Snomed OWL Toolkit or the Classification Service REST API which is language agnostic.

High Level Process

Classification

1. Read the Stated Form from RF2 files.

a. The following files are required: Concept, Stated Relationship, OWL Ontology Reference Set, OWL Axiom Reference Set and MRCM Attribute Domain Reference Set.

1. Use the OWL API to infer the class hierarchy

a. Build the Ontology object using:

i. Axioms from the OWL Axiom Reference Set, making a note of any Transitive property axioms.
ii. Axioms created by converting Stated Relationships to OWL Axioms using the MRCM Attribute Domain Reference Set for list of attributes which should not be grouped in the given domain.

a. Use a reasoner to pre-compute the class hierarchy.

Necessary Normal Form Calculation

Calculating the necessary normal form happens in two passes of the hierarchy.

1. Walk the class hierarchy in a top-down, breadth first, order.

a. For each class visited gather the stated attributes of this class and each inferred parent.
b. Compare the attributes and remove those which are found to be redundant because they are less specific in terms of depth in the hierarchy.
c. During this first pass build a hierarchy for property chains and transitive properties.

1. Walk the class hierarchy again in the same order reducing the attributes of each class further.

a. Compare the attributes and remove those which are found to be redundant because they are less specific in terms of depth in one of the alternate hierarchies.

For fine level detail the best source of information is the Java class org.snomed.otf.owltoolkit.normalform.RelationshipNormalFormGenerator which performs the Necessary Normal Form calculation.

3. Quality Assurance

The following additional quality assurance should be developed for the OWL refsets.

1. Validation of expressions following OWL Functional-Style Syntax specification
2. Validation of profile of the OWL ontology generated from the OWL refsets
3. Each active concept should have at least one active axiom
4. Each active concept can only have one declaration of Class, Object property, or Data property
5. Inactive concept must not have active axiom
6. Active axiom must not contain any inactive component
7. Domain and range validation by property types

1. All descendants of |Concept model object attribute| can only have target values from component of concept or expression
2. All descendants of |Concept model data attribute| can only have target values from data types and it must not have values from component of concept[1]

---

[1] Utilise the OWL API 4 or 5 for profile violation check as part of QA
http://owlcs.github.io/owlapi/apidocs_4/org/semanticweb/owlapi/profiles/OWLProfileViolation.html
http://owlcs.github.io/owlapi/apidocs_5/org/semanticweb/owlapi/profiles/OWLProfileViolation.html

4. OWL Expression Reference Set Specification

Purpose

An 762676003 |OWL expression type reference set| associates description logic statements with SNOMED CT concept in the OWL functional syntax.

The SNOMED CT International Release contains two reference sets that follow the 762676003 |OWL expression type reference set| pattern:

- The 733073007 |OWL axiom reference set (foundation metadata concept)|, in which the OWL expressions represent and axioms that form part of the definition of the concept identified by the referencedComponentId.
- The 762103008 |OWL ontology reference set (foundation metadata concept)|, in which the OWL expressions represent essential information about an ontology. This information includes, namespaces, ontology URI, ontology version URI, and import statements. The 762103008 |OWL ontology reference set (foundation metadata concept)| enables the use of prefixes in the ontology

Data Structure

An 762676003 |OWL expression type reference set| is structured as shown in the following table.

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T*.	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 |Module (core metadata concept)|within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 762676003 |OWL expression type reference set (foundation metadata concept)|	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. The concept to which the OWL expression applies. In the case of the 733073007 |OWL axiom reference set (foundation metadata concept)|, the axiom contributes to the definition of the identified concept.	NO	NO
owlExpression	String	The text of OWL expression to attach to the component identified by referencedComponentId.	YES	NO

Metadata

The following metadata supports this reference set:

900000000000454005 |Foundation metadata concept|
  900000000000455006 |Reference set|
    762676003 |OWL expression type reference set|
      762103008 |OWL ontology reference set|
      733073007 |OWL axiom reference set|
  900000000000457003 |Reference set attribute|
    706999006 |Expression|
      762677007 |OWL expression|
    900000000000459000 |Attribute type|
      900000000000465000 |String|
      762678002 |OWL 2 language syntax|

Descriptor Template and Examples

The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the descriptors that define the structure of the 762676003 |OWL expression type reference set| pattern and examples of the descriptors for specific reference sets that follow this pattern.

Table 4-1: Descriptor templates for OWL expression reference rets

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 | Reference set descriptor|	762676003 |OWL expression type reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0
900000000000456007 | Reference set descriptor|	762676003 |OWL expression type reference set|	762677007 |OWL expression|	762678002 |OWL 2 language syntax|	1
900000000000456007 | Reference set descriptor|	762103008 |OWL ontology reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0
900000000000456007 | Reference set descriptor|	762103008 |OWL ontology reference set|	762677007 |OWL expression|	762678002 |OWL 2 language syntax|	1
900000000000456007 | Reference set descriptor|	733073007 |OWL axiom reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0

900000000000456007 | Reference set descriptor|	733073007 |OWL axiom reference set|	762677007 |OWL expression|	762678002 |OWL 2 language syntax|	1

OWL Ontology Reference Set Example

Table 4-2: OWL ontology reference set example

moduleId	refsetId	referencedComponentId	owlExpression
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(:=<http://snomed.info/id/>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734147008 |OWL ontology header|	Ontology(<http://snomed.info/sct/900000000000207008>)

OWL Axiom Reference Set Example

Table 4-3: OWL axiom reference set example

moduleId	refsetId	referencedComponentId	owlExpression	Explanatory Notes
900000000000020 7008 |SNOMED CT core module|	733073007 |OWL axiom reference set|	404684003 | Clinical finding (finding)|	SubClassOf(:404684003 :138875005)	Example of SubClassOf, which is equivalent to an |Is a| relationship between most SNOMED CT concepts. ⚠ A different OWL expression is used to represent |Is a| relationships between attributes. This shown in the row below. • 404684003 |Clinical finding (finding)| • 138875005 |SNOMED CT Concept (SNOMED RT+CTV3)|

900000000000012004 |SNOMED CT model component module|	733073007 |OWL axiom reference set|	123005000 | Part of (attribute)|	SubObjectPropertyOf(:123005000 : 733928003)	Example of SubObjectPropertyOf, which is equivalent to an |Is a| relationship between attributes. • 123005000 |Part of (attribute)| • 733928003 |All or part of (attribute)|
900000000000207008 |SNOMED CT core module|	733073007 |OWL axiom reference set|	90708001 | Kidney disease (disorder)|	EquivalentClasses(:90708001 ObjectIntersectionOf(:64572001 ObjectSomeValuesFrom(:609096000 ObjectSomeValuesFrom(:363698007 : 64033007))))	Example of EquivalentClasses. which is equivalent to stating that a concept is sufficiently defined by relationships a set of defining relationships. • 90708001 |Kidney disease (disorder)| • 64572001 |Disease| • 609096000 |Role group (attribute)| • 363698007 |Finding site (attribute)| • 64033007 |Kidney structure (body structure)|
900000000000207008 |SNOMED CT core module|	733073007 |OWL axiom reference set|	126516008 | Neoplasm of skin of upper limb (disorder)|	EquivalentClasses(:126516008 ObjectIntersectionOf(:64572001 ObjectSomeValuesFrom(:609096000 ObjectIntersectionOf(ObjectSomeValuesFrom(:116676008 :108369006) ObjectSomeValuesFrom(:363698007 : 371311000)))))	Example of a role group with a conjunction of two relationships as its value. • 126516008 |Neoplasm of skin of upper limb (disorder)| • 64572001 |Disease| • 609096000 |Role group (attribute)| • 116676008 |Associated morphology (attribute)| • 108369006 |Neoplasm (morphologic abnormality)| • 363698007 |Finding site (attribute)| • 371311000 |Skin structure of upper limb (body structure)|
900000000000012004 |SNOMED CT model component module|	733073007 |OWL axiom reference set|	123005000 | Part of (attribute)|	TransitiveObjectProperty(:123005000)	Example of a transitive object property. • 123005000 |Part of (attribute)|
900000000000012004 |SNOMED CT model component module|	733073007 |OWL axiom reference set|	733930001 | Regional part of (attribute)|	SubObjectPropertyOf(ObjectPropertyChain(: 127489000 :738774007) :127489000))	Example of a property chain. • 127489000 |Has active ingredient (attribute)| • 738774007 |Is modification of (attribute)|

900000000000012004 |SNOMED CT model component module|	733073007 |OWL axiom reference set|	733929006 |General concept inclusion axiom|	EquivalentClasses(ObjectIntersectionOf(:244066003 ObjectSomeValuesFrom(:733930001 ObjectIntersectionOf(:244066003 ObjectSomeValuesFrom(:733931002 :302548004)))) ObjectIntersectionOf(:244066003 ObjectSomeValuesFrom(:733931002 ObjectSomeValuesFrom(:733930001 :302548004))))	Example of a general concept inclusion (GCI). • 733929006 |General concept inclusion axiom| • 244066003 |Entire skin region (body structure)| • 733930001 |Regional part of (attribute)| • 244066003 |Entire skin region (body structure)| • 733931002 |Constitutional part of (attribute)| • 302548004 |Entire head (body structure)|