Style Guide for Documentation of the CASE Ontology

Version 1.0

CASE Community Guidance Document

14 January 2020

Introduction

The Cyber-investigation Analysis Standard Expression (CASE) ontology is intended to be used to represent digital forensics information that may be shared among organizations cooperating in the conduct of digital investigations. This *Style Guide for Documentation of the CASE Ontology* provides guidance for creating metadata to be included in the CASE ontology using annotation properties associated with the ontology and its components (*i.e.*, classes, properties, and datatypes). This human-readable metadata ("annotations") is included in and published with the encoded ontology in order (1) to promote the intelligibility of the ontology as a model for the domain of cyber-investigation and (2) to support the interoperability of information encoded in accordance with the CASE standard.

The annotations of the ontology document the formal name, scope, purpose, and version information about the ontology. The annotations of the ontology components document the terminology and conceptual content of the ontology in a human-readable text format. Annotations define the designations (names, labels, and aliases) and definitions of the concepts represented in the ontology. The designations reflect the terminology of subject-matter experts (SMEs) in the domains of digital forensics and cybersecurity. The definitions are intended to capture experts' characterization of the domain concepts that are represented in the formal ontology. Initial CASE documentation is in the English language.

This Style Guide provides detailed guidelines for the formulation of designations, definitions, and other documentation of the ontology and ontology components. All official CASE ontology development shall follow the guidance provided in this Style Guide. Questions about the interpretation or application of the guidelines in this Style Guide, and requests for guidance on topics not covered by this document, should be addressed to the CASE Ontology Committee Coordinator or Ontology Committee Chair. Contact information is available on the CASE Community website (https://www.caseontology.org).

Throughout this document, the phrase "the ontology" is used to refer to the CASE ontology, which is managed for the CASE Community through the authority of the CASE Governance Committee. The CASE Ontology Committee is the community forum responsible for maintenance, development, and publication of the ontology. In that role, the Ontology Committee oversees prioritization of change requests, community coordination, technical development, and change notifications for updates and new versions of the ontology.

Revision History

Version	Approval Date	POC	Change Description
1.0	2020-01-14	CASE Ontology Committee Chair	Original publication.

Table of Contents

Table of Tables

1 Purpose

The *Style Guide for Documentation of the CASE Ontology* specifies guidance for documenting the ontology and its representational components (*i.e.*, classes, properties, and datatypes) with human-understandable annotations. This Style Guide provides detailed guidelines for the formulation of the names, definitions, and other annotations encoded with the ontology. The CASE Ontology Committee requires adherence to this guidance document for documentation in CASE ontology development efforts.

2 References

The documents listed in Table 1 are essential to the understanding of this document. For dated references, only the cited edition or version applies. For undated references, the latest edition or version of the referenced document (including any amendments) applies.

Table 1 – Normative References

Standard or Specification
OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition): http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/
OWL 2 Web Ontology Language: Mapping to RDF Graphs (Second Edition): http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/
RDF Schema 1.1: W3C Recommendation 24 February 2014: http://www.w3.org/TR/rdf-schema/
W3C rdf:PlainLiteral: A Datatype for RDF Plain Literals (Second Edition) (11 December 2012): http://www.w3.org/TR/2012/REC-rdf-plain-literal-20121211/
W3C SKOS Simple Knowledge Organization System (18 August 2009): http://www.w3.org/TR/2009/REC-skos-reference-20090818/
DCMI Metadata Terms (14 June 2012): http://dublincore.org/documents/dcmi-terms/
IETF RFC 3987, Internationalized Resource Identifiers (IRIs): http://www.ietf.org/rfc/rfc3987.txt
IETF RFC 5646, BCP 47, Tags for Identifying Languages (clause 2.2.1 (1))*:* http://www.ietf.org/rfc/bcp/bcp47.txt

The informative (non-normative) documents listed in **Table 2** are useful to understanding and using this document.

Table 2 – Informative References

Standard or Specification
OWL 2 Web Ontology Language: Quick Reference Guide (Second Edition): http://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/
IEEE Editorial Style Manual for Authors: http://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE_Style_Manual.pdf.
Merriam-Webster.com Dictionary. (Online) https://www.merriam-webster.com/ (“Merriam-Webster.com Dictionary”) NOTE: This online dictionary was originally based on the Eleventh Edition print version of their *Collegiate Dictionary*; however, the online content has since been updated and expanded.
Merriam-Webster’s Collegiate Dictionary, Eleventh Edition. (Print)

3 Terms, Definitions and Acronyms

3.1 Terms and Definitions

Terms and definitions[1] used in this document are presented in Table 3.

Table 3 – Definitions Applicable to this Document

Term	Definition
annotation	An expression used to associate information with an ontology or other resource, such as an ontology component. NOTE 1: An annotation is additional information associated to ontologies or ontology components that is intended for human consumption and not for use by reasoning software. NOTE 2: Each annotation consists of an annotation property and an annotation value. SOURCE: OWL 2 Structural Specification (Section 3.5; Section 10)
axiom	A statement of something that is true in the universe of discourse (domain). NOTE 1: Axioms in OWL 2 can be declarations, axioms about classes, axioms about object or data properties, datatype definitions, keys, assertions (sometimes also called facts), and axioms about annotations. SOURCE (NOTE 1): OWL 2 Structural Specification (Section 9) NOTE 2: A universe of discourse is a view of the real or hypothetical world that includes everything of interest. SOURCE (NOTE 2): ISO 19150-2 citing ISO 19101-1:2014 (Clause 4.1.38)
concept	A mental representation of knowledge as an abstraction of the essential characteristics of a type of entity, or a relationship between entities, in a subject area or domain. NOTE: The semantics (meaning) of a concept may be represented formally using a logically based knowledge representation language. SOURCE: *The Semantic Web*. Michael C. Daconta, Leo J. Obrst, Kevin T. Smith. 2003.
controlled vocabulary	A set of **vocabulary term**s consisting of defined **lexical item**s (*i.e.*, words, phrases, or abbreviations from a natural language) that are collected and managed by an authority following identified criteria for inclusion.
IRI	A sequence of characters from the Universal Character Set (Unicode/ISO 10646) [IETF RFC 3987], intended for identifying an abstract or physical resource. NOTE 1: Every URI is by definition an IRI. A mapping from IRIs to URIs is defined, which means that IRIs can be used instead of URIs, where appropriate, to identify resources. SOURCE (NOTE 1): IETF RFC 3987 NOTE 2: A resource can be anything that has identity, *e.g.*, an OWL class instance and its associated annotations. NOTE 3: OWL 2 extends OWL 1 by specifying the use of IRIs to identify ontologies and their components. OWL 1 uses Uniform Resource Identifiers (URIs). SOURCE (NOTE 3): OWL 2 Structural Specification (Section 2.4)
lexical item	A word, phrase, or abbreviation represented as a character string that expresses content in a specified natural language.
namespace	In RDF, a common URI prefix or stem (including a **URI base** plus a terminal separator) used in identifiers for a set of related resources. NOTE 1: The RDF namespace is concatenated with a local name to create the complete URI identifier for an RDF resource. NOTE 2: Every RDF resource is identified by a URI. SOURCE (NOTE 2): ISO 19150-2:2015

[1] In the definitions, a term is styled in **bold** when the meaning of that term is specified elsewhere in Section 3.1

	NOTE 3: The CASE ontology uses the standard prefix names for namespaces as declared in the OWL Structural Specification (Section 2.4).
ontology component category	A classification of an ontology construct (examples: entity class, property) to indicate whether that construct represents a **concept** that describes a class of entities, a property (data property or object property), a class of values (*i.e.*, enumeration), or an individual listed value (*i.e.*, enumerant). NOTE: See Annex A.
URI base	A base URI in a domain owned by the organization that maintains the model or ontology. SOURCE: ISO 19150-2 (Clause 6.2.2).
Uniform Resource Identifier (URI)	A compact string of characters for identifying an abstract or physical resource. NOTE 1: A resource can be anything that has identity, *e.g.*, an OWL class instance and its associated annotations. NOTE 2: A URI identifies a resource either by location, or by name, or both. NOTE 3: URIs are limited to a subset of the ASCII character set. SOURCE: IETF RFC 3986
Uniform Resource Locator (URL)	A compact string representation for location and access of a resource available on the internet. NOTE: A URL is a type of URI. SOURCE: IETF RFC 1738
vocabulary term	A defined **lexical item** that represents a **concept** that describes real-world phenomena. EXAMPLE: A vocabulary term "Event" defined as "A temporal entity consisting of change over time (whether extended or instantaneous) involving (simple or complex) interaction(s) of physical entities and which may have geometric position and extent."

3.2 Acronyms

The acronyms that are used in this document are specified in the following list.

ASCII American Standard Code for Information Interchange
CASE Cyber-investigation Analysis Standard Expression
IRI Internationalized Resource Identifier
ISO International Organization for Standardization
OWL Web Ontology Language
RDF Resource Description Language
RDFS RDF Schema
SKOS Simple Knowledge Organization System
UCO Unified Cyber Ontology
URI Uniform Resource Identifier
URL Uniform Resource Locator
W3C World Wide Web Consortium
XML Extensible Markup Language
XSD XML Schema

4 Identifiers (IRIs) for the CASE Ontology and Ontology Components

Each representational component (*e.g.*, class, property, or enumeration) in the ontology has a unique identifier in the form of an IRI. The structure of these IRIs is described below.[2] IRIs for ontology components terminate in string segments (see **concepts** and **sub-concepts** below) that are unique within the ontology namespace. In the CASE ontology, these terminal segments typically share string

[2] A future edition of this document will address the placement and format of IRI segments to indicate a version of a resource.

elements with the human-readable names for ontology components. It is therefore relevant to describe the specification format for these IRIs.

IRI designations for CASE ontology resources[3] are constructed in accordance with the following pattern:

protocol '://' domain '/' resource-type '/' resource '/' sub-resource '#' concept ['.' sub-concept]

The named segments in the CASE IRI pattern are replaced by case-sensitive strings, determined as follows:

- **protocol** – always 'https'
- **domain** – always 'caseontology.org'
- **resource-type** – the type of resource (*e.g.*, 'ontology')
- **resource** – always a designator for a particular resource of the indicated resource-type (*e.g.*, 'case')
- **sub-resource** – always a designator for a particular resource that belongs to the main resource and is of the same resource-type (*e.g.*, 'core')
- **concept** – the designator for a particular ontology component within the resource (*e.g.*, the classes 'Hash', 'Action', 'ComputerSpecification', or 'CompilerType'; or the properties 'registryValue' or 'parentProcess')
- **sub-concept** – optional designator for an ontology component that is a named individual representing a listed value (*i.e.*, enumerant) of an enumerated type (*i.e.*, enumeration). For example, 'EncodingMethod.Base32' as the designator for an enumerant in the enumeration 'EncodingMethod'.[4]

Conventions for formulating the string for a **concept** segment of a CASE IRI depend upon the type of ontology component being identified. The following four types of ontology component are distinguished as a basis for naming conventions. See Annex A for a description of the types:

1. Class
2. Property
3. Enumeration (or other class representing a set of data values)
4. Enumerant (listed value or enumerated value).

Conventions for strings used in the IRIs to designate concepts depend on the type of ontology component:

1) All **concept** (and **sub-concept**) strings shall use camel-case (*i.e.*, shall be strings that contain no spaces and in which any string consisting of multiple words shall capitalize each word after the first word in the string).

2) A Class **concept** string shall begin with an upper-case letter (*i.e.*, shall be in upper camel-case).

[3] CASE ontology resources include the CASE ontology as a whole, its sub-ontologies, and individual concepts belonging to the CASE ontology. Following the RDF specification, anything in the world (*i.e.*, the "universe of discourse" for the domain of interest) that is denoted by an IRI or literal is called a ***resource***: "Anything can be a resource, including physical things, documents, abstract concepts, numbers and strings.... The resource denoted by an IRI is called its *referent*, and the resource denoted by a literal is called its *literal value*." RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. 25 February 2014 (http://www.w3.org/TR/rdf11-concepts/).

[4] The CASE ontology-design document will address CASE best practices for representing enumerants as named individuals or as literals.

3) A Class **concept** string shall be a singular noun, unless each individual member of the class is itself a plural entity.

4) A Property **concept** string shall begin with a lower-case letter (*i.e.*, shall be in lower camel-case).

5) Property **concept** strings are recommended to use strings that indicate the directionality of the relationship (*e.g.*, 'partOf' or 'hasPart', rather than only 'part').

6) An Enumeration **concept** string shall be a singular noun that begins with an upper-case letter (*i.e.*, shall be in upper camel-case).

7) An Enumerant **sub-concept** string shall be a singular noun that begins with an upper-case letter (*i.e.*, shall be in upper camel-case); it shall be used following the **concept** string for the Enumeration to which it belongs, with the '.' separator.

8) Recommended practice is that **concept** strings should be unique in spelling independently of case; in other words, it is recommended that an ontology *not* have both 'hash' and 'Hash' as **concept** strings.

The IRI **concept** string (or, if applicable, **sub-concept** string) is also used in the annotation property *label* (implemented using rdfs:label), as discussed in Section 7.4.

5 Documentation of the CASE Ontology

The CASE ontology is specified using the Web Ontology Language OWL 2. In addition to external documentation, an OWL 2 ontology may be described using formal annotations, which are human-readable metadata included with the encoded ontology. Annotations are linked to the ontology or ontology components using annotation properties. The OWL 2 language includes a set of annotation properties that can be used to provide information about an ontology.[5] Other RDF-based standards also define annotation properties that may be used with RDF resources (which includes ontologies and ontology components). Those standards include RDFS (with annotations, *e.g.*, rdfs:label, rdfs:comment), SKOS (*e.g.*, skos:altLabel), Dublin Core (*e.g.*, dct:source), and others.[6] Annotation properties allow metadata statements to exist within the ontology encoding rather than as file comments which are discarded when loading the ontology into a development tool or knowledge base.

Annotations are used to represent the human-readable metadata about the ontology and ontology components. The next section (Section 6) lists the types of annotations recommended for documenting the CASE ontology and its components. The minimal required documentation for a CASE ontology component is (1) a *label* and (2) a *definition*. Section 7 provides detailed style guidance for the creation of annotations in the CASE ontology. All documentation should be clear and comprehensible. Style guidance may be specialized based on the category of ontology component, *i.e.*, Entity Class, Property, Enumeration, or Enumerant (see Annex A). The spelling conventions followed

[5] OWL 2 Structural Specification (11 December 2012), Sections 3.1, 3.5, 5.5, and 10. Available online at http://www.w3.org/TR/owl2-syntax/. For encoding, also see OWL 2 Mapping to RDF Graphs (11 December 2012), available online at http://www.w3.org/TR/owl2-mapping-to-rdf/. Note: Annotation properties are not part of the logical content of an ontology, and they are not used in machine reasoning.

[6] Annotation properties may be related by subproperty axioms; for example, certain specialized annotations in the SKOS standard (skos:prefLabel, skos:altLabel) are defined as subproperties of rdfs:label.

should be those of the *Merriam-Webster.com Dictionary* (first spelling), unless otherwise approved by the CASE Ontology Committee. Section 8 contains encoding tables that specify the annotation properties used to formalize the annotations in the serialization of the ontology.

In this document, "the ontology" refers to the CASE ontology, including its sub-ontologies. Other ontologies imported by CASE, notably the Unified Cyber Ontology (UCO), will include annotations belonging to the imported ontology. The style of documentation in an imported ontology may differ from the CASE guidelines.

6 Annotations for the Ontology and Ontology Components

For the CASE ontology, annotations are used to document the metadata listed below. The purpose of each type of annotation is described in general terms. The mandatory, conditional, or optional status of each annotation type is discussed in the following section, along with specific guidance for writing the documentation. Concrete syntax for the annotations is specified in the encoding tables in Section 8.

6.1 Annotations for the Ontology

The following types of annotations are used to document the ontology (*i.e.*, an OWL ontology resource as a whole), including the CASE sub-ontologies:

- *ontology IRI* – The unique sequence of characters, constructed in accordance with the syntax for Internationalized Resource Identifiers (RFC 3987), that is used to identify the ontology as a resource.
- *title* – The preferred human-readable lexical item (*i.e.,* word, phrase, or abbreviation) that is used to designate the ontology in a specified natural language; the name of the ontology.
- *comment* – A human-readable statement describing the scope and purpose of the ontology. NOTE: The suggested namespace abbreviation for the ontology may also be documented in this annotation (*e.g.*, "case-core" for the Core module of CASE).[7]
- *versionIRI* – The unique sequence of characters in IRI syntax that is used to identify a particular version of the ontology.
- *versionInfo* – A string providing version-control information about a particular version of the ontology (*e.g.*, "0.2.0" to indicate version 0.2.0).
- *priorVersion* – A prior version (specified by IRI) of the annotated ontology.
- *backwardCompatibleWith* – A prior version (specified by IRI) of the annotated ontology that is compatible with the current version of the ontology.
- *incompatibleWith* – A prior version (specified by IRI) of the annotated ontology that is incompatible with the current version of the ontology.

[7] Although the namespace abbreviation may be documented in the *comment*, the namespace abbreviation is encoded for use within the ontology resource file using @prefix or xmlns:prefix. For example, in the line "@prefix case-core: https://caseontology.org/ontology/case/core#" in CASE Core (Turtle file).

6.2 Annotations for Ontology Components

The following types of annotations are used to document ontology components (*i.e.*, resources contained in the ontology, such as classes and properties, that represent concepts in the domain of the ontology):

- *concept IRI* - The unique sequence of characters, constructed in accordance with RFC 3987, that is used to identify the ontology component (*e.g.*, Entity Class or Property) and that shares the base IRI of the *ontology IRI*.
- *label* – A human-readable but compressed (*i.e.,* camel-cased with no whitespaces) identifier for a concept, which is unique within the context of the ontology.
- *name* – The preferred human-readable lexical item (*i.e.,* word, phrase, or abbreviation) that is used to designate the concept represented by an ontology component within the ontology and in a specified natural language.
- *alias* – A functionally equivalent synonym for designating the concept represented by an ontology component, in an alternative context and/or in another natural language.
- *definition* – A precise, human-readable statement of the nature and normative properties of the concept represented by an ontology component.
- *description* – Information supplemental to the definition of a concept, consisting of one or more statements about its non-essential qualities, variations, scope, or context of application. For an ontology component representing a property, the definition may state the type of entity to which the property applies (*i.e.*, domain) and the type of its allowable values (*i.e.*, range), including (if applicable) the expected physical quantity for allowable values.
- *example* – A resource (*e.g.*, a plain text statement or a linked document) that illustrates the use of a concept.

7 Documentation Guidance by Type of Annotation (and Component)

7.1 Title (Ontology)

The *title* annotation is used to record the human-readable name of the ontology. If there is an abbreviation for the ontology, it may be included in parentheses after the formal name. For example: “Cyber-investigation Analysis Standard Expression (CASE)”.

7.2 Comment (Ontology)

The *comment* annotation is used to record human-readable information about the purpose and scope of the ontology.

7.3 Name (Ontology Component)

The *name* annotation is used to record the preferred human-readable lexical item (*i.e.*, word, phrase, or abbreviation in a natural language) that designates the concept represented by the ontology component. The natural language of the name shall be indicated. An exception is made for the names

of Enumerants represented as a string specified within an enumeration of literals[8]; in that case, the string is the name (it cannot be annotated with the *name* annotation property). The guidance below for formulating names of Enumerants is required, with the exception that strings representing enumerated literals may omit the language indicator.

Specification of the *name* of an ontology component shall follow the guidance below.

> EXAMPLE: The *name* "Investigative Action", for the class used to represent forensic examination activities that are part of a cyber-investigation.

Specifications of *names* within the CASE ontology shall follow the guidance below:

a) Each ontology component shall have a *name*. The *name* is **required**.

b) There shall be **exactly one** *name* for each ontology component.[9]

c) The **language** of the name shall be identified using the IANA language subtag code[10] for the language (for example: "en" for English).

d) The *name* shall be **unambiguous.**

e) The *name* shall be **unique** within the ontology where the component is defined:

   1) It is strongly recommended that the uniqueness of a *name* not be dependent solely upon the case of its component characters.

   2) The uniqueness of a *name* for an Enumerant may be established by including a reference to its Enumeration. For example: "MD5 (Hash Method)", "Medium (Confidence Level)".

f) The *name* shall be **singular** in grammatical form. For example: "Hash" rather than "Hashes".

   1) The only exception to this rule is for a *name* of a component representing a concept that is inherently plural, *e.g.*, where each instance of a class would itself be plural.

g) The *name* should be **short** (preferably less than 50 characters in length).

h) The *name*, if it contains multiple words, shall have **spaces** between words.

i) **Capitalization**:

   1) All words in the *name* shall be capitalized, except for articles and prepositions of less than eight characters in length.

   2) The *name* may be based on the *label* of the ontology component reformatted into the above style with spaces between words.

[8] See Annex A.
[9] Alternative designations shall be specified as *aliases*; guidance for *aliases* is specified in Section 7.3 of this Style Guide.
[10] IANA language subtag codes are the lowercase two-character codes contained in the Language Subtag registry administered by the Internet Assigned Numbers Authority (IANA) in accordance with the Internet Engineering Task Force (IETF) Recommendation for Comment (RFC) 5646. The language codes in the IANA Language Subtag registry are based on the International Organization for Standardization (ISO) 639 series of standards. The complete registry content is available at the following URL: http://www.iana.org/assignments/language-subtag-registry/language-subtag-registry.

i. For ontologies in which some *labels* are unique only based on capitalization, the *names* may be disambiguated by indicating the component type following the name; for example: the *name* "Hash" for the class with the *label* "Hash", and the *name* "Hash (Property)" for the property with the *label* "hash". Best practice is not to have identifiers whose distinction depends solely on case-sensitivity.

j) The *name* should follow consistent rules for **hyphenation.** The name portion following a hyphen should begin with a lower-case letter (for example: "Cyber-investigation").

1) An exception may be made for a *name* that is the expansion of an acronym (for example: "Transporter-Erector-Launcher").

k) The *name* shall not include **specialized terminology** unless that terminology is defined in the definition (*definitionNote*).

l) The *name* shall exclude **possessive pronouns** and adjectives (for example: "his", "her", "their", and "its").

m) The *name* shall not include special **symbols** (*i.e.,* a name shall use Basic Latin Unicode characters (ASCII)).

n) The *name* should not be an **abbreviation** or an **acronym**.

1) An exception is made for concepts that are more commonly designated by an abbreviation or acronym than by the expansion. In those cases, the *definitionNote* may contain the expansion and/or explanation of the abbreviation. For example, the compression method "JPEG" (an image compression format, named using the acronym for its creators, the "Joint Photographic Experts Group").

2) An exception is made for references to external authoritative standards that use abbreviations or acronyms in names of ontology components, as described below in subsection (p).

o) The *name* may include an **acronym** in parentheses following the full name (for example: "Aerodrome Reference Point (ARP)").

1) It is recommended that acronyms used as alternative designators for a concept be documented in separate annotations (see Section 7.3, "Aliases").

p) For all ontology components based on an **external authoritative standard** (and not imported from an external ontology), the *name* may use a community identifier included parenthetically at the end of the *name* to indicate the source. For example: "Image (SWGDE)".

1) This guideline applies to concepts for which ontology components are created based on an external standard whose content has not been represented in an ontology that could be imported.

q) The *name* of an Enumeration should end with the word "Type". For example: "File System Type".

1) An exception may be made for an alternative ending word if it is a more understandable indicator of a set of values for a specific topic, *e.g.*, "Action Status" or "Compression Method".

r) The *name* of each Enumerant may append, in parentheses, the *name* of its associated Enumeration, for clarity. For example: "Public (Visibility Type)".[11]

1) The *names* of Enumerants are unique within the context of their Enumeration.

2) The *name* of an Enumerant that is a literal (rather than a named individual) is a string representing a listed value within an enumeration of literals.

7.4 Alias (Ontology Component)

An *alias* annotation is used to record a functionally equivalent synonym for designating a concept in an alternative context and/or another language where an alternative to the *name* is more familiar or useful. Acronyms and abbreviations may be documented as *aliases*.

Specifications of *aliases* for ontology components shall follow the guidance below:

EXAMPLE: The *alias* "Photographic Comparison" for the concept *named* "Image Comparison"

a) An ontology component is not required to have an *alias*. *Aliases* are **optional**.

b) An ontology component may have **multiple** *aliases*.

c) The **language** of each *alias* shall be identified using the IANA language subtag code for the language (for example: "en" for English).

d) An *alias* need not be **unique** within the vocabulary where the term is defined.

e) An *alias* should be **singular** in grammatical form. For example: "Property Bundle" rather than "Property Bundles". An exception may be made for a concept that is inherently plural or for a common usage.

f) An *alias* should use consistent rules for **hyphenation.** The term following a hyphen should begin with a lower-case letter (for example: "Cyber-investigation").

1) An exception may be made for an *alias* that is the expansion of an acronym (for example: "Transporter-Erector-Launcher").

2) An exception may be made for an *alias* that reflects the punctuation rules of its alternative context or language.

g) An *alias* shall exclude **possessive pronouns** and adjectives (for example: "his", "her", "their", and "its").

h) An *alias* shall not include special **symbols** (*i.e.,* an a*lias* shall use Basic Latin Unicode characters (ASCII)).

[11] This option applies only to Enumerants belonging to an Enumeration of Individuals. Enumerants in Enumerations of Literals do not have *name* annotations.

i) An *alias* may be an **acronym** or an **abbreviation**.

j) An *alias* may include a phrase parenthetically at the end of the *alias* to identify the community or context in which the alias is used.

k) An *alias* for an Enumeration should end with the word "Type" or an alternative indicator of a set of values. For example: "Disk Type", "Action Status".

l) An *alias* for an Enumerant may append, in parentheses, the *name* of its associated Enumeration, for clarity.[12]

7.5 Label (Ontology Component)

The *label* annotation is used to record a human-understandable but compressed (*i.e.,* camel-cased) identifier that is unique within the ontology that contains the component.

Specifications of *labels* within the CASE ontology shall follow the guidance below.

EXAMPLE: The *label* "HashMethod" (for the ontology component with the *name* "Hash Method")

a) Each ontology component shall have a *label*. The *label* is **mandatory.**

b) The *label* string shall correspond to the terminal segment of the IRI identifier of the concept in the CASE ontology (see Section 4).

c) There shall be **exactly one** *label* for each ontology component.

d) The **language** of the *label* shall be identified using the IANA language subtag code for the language (for example: "en" for English).

e) The *label* shall be **unambiguous**.

f) The *label* shall be **unique** within the ontology where the ontology component is defined, as follows:

    1) It is strongly recommended that the uniqueness of a label not be dependent upon the case of its component characters.

g) A *label* should be a camel-cased rendition of the *name* annotation string, applying the following rules for upper or lower camel-case (with an exception for acronym strings used in a *label* (see (k) below), which retain their original capitalization style):

    1) For **camel-case**, remove any whitespaces or hyphens that occur within the *name* string, and capitalize the initial letter of each word or word-segment that follows a removed whitespace or hyphen. A single-word *name* will not be affected by this process; however, the following upper or lower camel-case rules shall be applied to its first letter.

[12] This option applies only to Enumerants belonging to an Enumeration of Individuals. Enumerants in Enumerations of Literals do not have *alias* annotations.

2) **Upper camel-case** (applies to the *label* for an Entity Class, Enumeration, or Enumerant): The *label* begins with an upper-case (capital) letter. For example: "HashMethod" (for the Entity Class with the *name* “Hash Method”).

    i. For an Enumerant with the Enumeration *name* included as context, the upper camel-case rule applies to the part of the *label*, following the period, that specifically names the listed value. For example: “AccessibilityStatusTermSet.Closed)” for the listed value with the *name* “Closed (Accessibility Status)”. See 7.2 (r) on *name* for an Enumerant.

3) **Lower camel-case** (applies to the *label* for a Property): The *label* begins with a lower-case letter. For example: “registryValue” (for the property with the *name* “Registry Value”).

h) The *label* for an Enumeration shall end with the string “Type” (or an alternative indicator of a set of values). Examples: “FileSystemType” and (alternative indicator) “ActionStatus”.

i) The *label* for each Enumerant shall be in upper camel-case.

1) The *label* of an Enumerant shall be unique within the context of its Enumeration.

2) The *label* of an Enumerant may consist of the concatenation of the *label* of its associated Enumeration followed by a dot (“.”) and an upper camel-cased string unique to the Enumerant. For example: “AccessibilityStatusTermSet.Closed”. All Enumerants in the same Enumeration shall follow the same style.

3) (Alternative to (2)) The *label* of an Enumerant may consist of the concatenation of the literal with an underscore (‘_’) and the Enumeration *label*. For example: “Closed_AccessibilityStatusTermSet”. All Enumerants in the same Enumeration shall follow the same style.

j) The *label* should be short (preferably less than 50 characters in length).

k) For conciseness, a *label* may use acronyms and abbreviations, if appropriate. Abbreviations should consist of intelligible substrings. An acronym shall retain its original capitalization style when used in a *label*; this is an exception to the camel-case specification above in (g).

l) For ontology components which are based on an **external authoritative standard** (*e.g.*, ISO 3166-1:2013), but not imported from an external ontology, the *label* may include a community identifier at the end, following an underscore. For example: “countryName_Iso3166-1-2013” for the *label* of the property whose value is a country name from the ISO 3166-1 (2013) standard for country names and codes. Note: Including an identifier in the *label* is **optional**; the basis of an ontology component may instead be recorded using the *definitionNote* or alternative annotation (*e.g.*, dct:source) on the component.

7.6 Definition (Ontology Component)

The *definition* annotation is used to record a precise statement of the meaning of the concept represented by the ontology component.[13] The definition identifies the essential characteristics of the concept represented by the ontology component.

EXAMPLE 1: (*definition* for "Event" (an Entity Class)) "An event is a temporal entity consisting of one or more changes over time (whether extended or instantaneous), involving simple or complex interaction(s) of physical entities, and which may have geospatial position and extent."

EXAMPLE 2: (*definition* for "Spatial Unit" (an Entity Class)) "A spatial unit is an area of land, water, and/or a volume of space, which has been described (legally or informally) for the purpose of recognizing rights, restrictions, or responsibilities."

EXAMPLE 3: (*definition* for "Horizontal Clearance" (a Property)) "The horizontal clearance of a transitway is the distance available to pass a load that extends laterally beyond the wheels of a vehicle."

The *definition* for a CASE ontology component shall follow the guidance below. Spelling shall conform to the *Merriam-Webster.com Dictionary* (online).

a) Each ontology component (*i.e.*, class, property, enumeration, and listed value) shall have a *definition*. The *definition* is **mandatory**.

b) There shall be **exactly one** *definition* for each ontology component.

c) The **language** of the *definition* shall be identified using the IANA language subtag code for the language (*e.g.*, "en" for English).

d) A *definition* is **a succinct statement of the essential meaning** of an ontology component in language understandable to a human user. The *definition* should indicate the relevant general concept (if any) together with the unique necessary characteristic(s) that distinguish the concept being defined from the general concept (see Example below). A *definition* will rely on general background knowledge about the world and the types of objects and relationships in it.

[13] Annotations for recording additional information beyond the essential definition are described in later sections. See Section 8.3 (for a table of all CASE annotation properties) and Annex B (for other specialized annotation properties).

EXAMPLE 4: The ontology classes "Building" and "Non-building Structure" both refer to types of constructions, with specific differences identified in their definitions (*e.g.*, buildings are designed for human occupancy, while non-building structures are not). In these examples, examples are embedded to explain both the general and specific concepts.

Building: "A building is a free-standing, self-supporting construction that is roofed, usually walled, and intended for human occupancy (for example: an office, a factory, or a sports stadium) and/or habitation (for example: a house, apartment building, barracks, or nursing home)."

Non-building Structure: "A non-building structure is a free-standing, self-supporting construction (for example: a large piece of equipment) designed to support human activities (for example: agriculture, manufacturing, or mining) but not intended for human occupancy and/or habitation."

e) The *definition* shall consist of **exactly one sentence** stating the essential meaning of the concept being defined.

   1) (Best Practice) The *definition* sentence should be a complete sentence having both a subject and a predicate (*e.g.*, "A building is a free-standing, self-supporting construction that is roofed, usually walled, and intended for human occupancy....").

   2) The *definition* sentence may optionally be a sentence fragment in which the subject is understood to be the name of the ontology component being defined and only the predicate is expressed (*e.g.*, "A free-standing, self-supporting construction that is roofed, usually walled, and intended for human occupancy ...."). This form has a potential disadvantage if the documentation is used separately from the ontology (*e.g.*, in autogenerated HTML documents or as rollover text in a GUI), because the sentence fragment may be separated from the subject of the sentence.

f) When used in the *definition*, the name of the ontology component should not be capitalized unless it is a proper noun in its natural language.

g) The *definition* should contain only **one independent clause** (a subject and predicate), optionally with dependent clauses.

h) The *definition* shall not use word(s) that are used in the concept name in the definition without defining them (*i.e.*, no circular definitions).[14]

i) The *definition* shall not include **idioms** or idiomatic expressions (for example: "hands down", "at the drop of a hat").

j) The *definition* shall be grammatically **singular** in agreement with the name.

k) **Acronyms** used in the *definition* shall be spelled out on first use unless they are in widespread use and commonly understood (*e.g.*, "USA").

[14] See IEEE-SA Standards Style Manual, Annex B.1: "Terms themselves should not be used in their own definitions." Such definitions are commonly called "circular definitions" because they are useful only to those who already know what the term being defined means.

l) The *definition* may include an example of a concept being used to define the main concept (see EXAMPLE 4, above, and EXAMPLE 5, below); however, examples shall not be used in place of the definition for the main concept.

1) If necessary, the *definition* may include an example of a critical term or phrase that is used in the definition, in the form "(for example: ...)".

2) Examples of the concept being defined (if any), should be documented using an *example* annotation (Section 7.6).

EXAMPLE 5: (Definition for "Street Sign", with an example to illuminate a critical phrase) "A sign placed along a road for the purpose of regulating the flow of traffic (for example: moving vehicles or pedestrians) and/or providing information."

m) The Definition shall conform to the following guidance for **punctuation**:

1) The *definition* shall begin with a capital letter and end with a period.

2) The *definition* shall not capitalize words unless they are proper nouns or capitalized acronyms.

3) The *definition* shall use the **Oxford comma** (also called Harvard comma, serial comma, or series comma) punctuation. That is, a comma shall be used after all but the final item in a series, including the penultimate item preceding the coordinate conjunction (usually "and" or "or").

4) The *definition* may include semicolons, colons, and other applicable punctuation.

5) Alternation:

a. The English "or" indicates an inclusive alternation (*i.e.*, "A or B", where one or both of A and B may apply).

b. The phrase "**and/or**" may be used to *emphasize* an inclusive alternation.

c. To indicate an exclusive alternation, use "either" and "or" (*i.e.*, "either A or B", where only one of A and B may apply).

6) Representation of **numbers** follows the U.S. convention, in which the period ('.') is used as the radix marker (the decimal point), and the comma (',') is used to delimit groups of three digits to the left of the radix marker.

7) The *definition* should spell out **units-of-measure** (for example: "kilometers per hour", rather than "km/h") if needed for clarity.

n) The *definition* should specify the intended **denotation** of an ontology component (*i.e.*, the item denoted in the universe of discourse).

o) The *definition* of a non-imported ontology component that is based on an **external authoritative standard** should include a reference to the external standard. For example: (for the *definition* of "Classification Code" based on the ISO Metadata standard) "The set of the named levels of handling restrictions that may be applied to a resource. (ISO 19115-1:2014)"

1) This approach should be used when the content of the external standard is not available as a formal ontology.

7.7 Description (Ontology Component)

The *description* annotation is used to record additional explanation about the meaning of an ontology component, including non-essential characteristics, variations, context, scope, or usage. A *description* for certain kinds of concepts (*e.g.*, properties for measurements) may include specialized information (*e.g.*, expected physical quantity or unit of measure).

The *description* shall follow the guidance below. Spelling shall conform to the *Merriam-Webster.com Dictionary* (online).

a) The *description* annotation contains an **elaboration on the meaning** of the concept specified in the *definition*.

b) The *description* annotation is **optional**.

c) A *description* shall consist of at least one **complete sentence**.

d) There **may be more than one** sentence in a *description*.

e) A *description* may contain complex sentences with **multiple clauses**.

f) A *description* may include examples of the concept, following the phrase “For example: ” or “(for example:)”, when immediately after a term/phrase; however, use of the *example* annotation (Section 7.8) is preferred.

EXAMPLE 1: (*description* for "Event" (an Event)): "An event involves one or more changes in physical entities. An event has temporal properties (for example: date(s) and duration) and may have temporal parts. An instantaneous event is the limit case (in time). Temporal properties are used to describe the duration of an event in time; for example, specified date(s) of a calendar time period and/or a specified duration of the event. An event may have point (instantaneous) or interval (extended) duration. Events may have temporal parts which are also events. Temporal parts of an event may be sequential (for example: innings of a baseball game) or concurrent (for example: sub-events of a complex event, which are occurring at the same time, such as voting activities at many separate polling places during a national election). Events may be long-duration processes (for example: the construction of Hoover Dam), with continuous or discontinuous activity, or discrete events which have continuous activity bounded by definite date-times, whether known or unknown (for example: running the Boston Marathon; the lifetime of Alexander the Great). An event may have geospatial position and extent based on the location(s) where the interactions occur and the spatial extent of the actors and effects involved."

EXAMPLE 2: (*description* for "Spatial Unit" (an Entity Class)) "A spatial unit may be described by official cadastral data or by an informal method (for example: a social-tenure tradition referring to landmarks). A spatial unit may represent a geospatial region that is not otherwise classified as a specific type of physical feature. The intended application is in accordance with ISO 19152:2012 (Land Administration Domain Model)."

EXAMPLE 3: (*description* for "Horizontal Clearance" (a Property)) "Value Type: Real Interval (non-negative). Physical Quantity: Length."

g) **Acronyms** shall be spelled out on first use unless previously expanded in the *definition* or unless they are in widespread use and commonly understood (*e.g.*, "USA").

h) A *description* shall follow the **punctuation** rules specified above in 7.5.1(m) for the *definition*.

7.7.1 Value Type

The *description* for an ontology component that is a Property shall **conditionally** include a special part (**Value Type**) following the guidance below:

a) The **Value Type** part of a *description* shall indicate the set of allowed values for the Property.

   1) The Value Type is **mandatory** for a Property whose allowed values are specified by a set of quantitative values (*e.g.*, the non-negative integers).

   2) For mandatory cases, there shall be **exactly one Value Type** specified for the Property.

   3) The Value Type part is **optional** for other kinds of Property.[15]

b) The Value Type shall be indicated in a phrase of the form "Value Type:" followed by the type (*e.g.*, (for the Property *stackSize*) "Value Type: Non-negative Integer"), which is included in the *description*.

[15] Alternatively, the type of allowed values for a Property may be indicated within its *definition* (e.g., (for the Property "Created By") "The relationship between an object in the domain and the identity (*i.e.*, individual or organization) that made it.").

c) The Value Type is expressed as a **text string** that denotes the datatype or class specifying the allowed value(s) of the Property. For example: "Non-negative Integer", “String”, "Boolean", “Account Type”, or “Identifier”. It is not necessary to cite a standard for the Value Type unless there is some potential for confusion otherwise.

7.7.2 Physical Quantity

The *description* for an ontology component that is a Property shall **conditionally** include a special part (**Physical Quantity**) following the guidance below:

a) The *description* for a Property whose Value Type represents a measurement of a physical quantity shall indicate the reference unit of the Physical Quantity for allowed values of the Property.

1) The Physical Quantity is **mandatory** for a Property whose value is the magnitude of a physical-quantity measurement.

2) The *description* for such a Property shall include **exactly one** part indicating the expected reference unit of the Physical Quantity.

b) The Physical Quantity indication shall follow the Value Type and be prefaced by “Physical Quantity:” (*e.g.*, (for the Property *stackSize*) “Value Type: Non-negative Integer. Physical Quantity: Byte”).

c) The Physical Quantity is expressed as a **text string** specifying the expected reference unit for allowed values for the Property. It is not necessary to cite a standard for the Physical Quantity unless there is some potential for confusion otherwise.

7.8 Example (Ontology Component)

The *example* annotation is used to record an illustration of the use of an ontology component. An example may be described in plain text or in a linked resource containing a document, diagram, and/or sample encoding.[16] An example shall contain a particular instance of a concept in a context where the meaning is clear.

[16] For flexibility of *example* annotations, see the *SKOS Reference* (18 August 2009), Section 7.1 (http://www.w3.org/TR/skos-reference/#notes).

8 Encodings for Annotations

8.1 Format of Encoding Tables

Encoding elements used for annotations in the CASE ontology are specified in the tables in this section. The table format used to document these encoding elements is as follows:

- The first column contains a row **Reference** number.
- The **Type of Documentation** column identifies the kind of information recorded by the property.
- The **Multiplicity** column indicates the number of occurrences of the annotation property permitted by these guidelines.
- The **Value Type** column indicates the datatype of the value of the annotation property. Datatypes may be referenced from a standard (*e.g.*, XML datatypes) or defined within the ontology itself (*e.g.*, an enumeration created as a value type for a property defined in a specialized domain).
- The **Notes** column contains comments on the purpose of this annotation in the CASE ontology, optionally with example values. Any restrictions on the use are also noted in this column.
- The **Annotation Property** column specifies the element (prefixed with standard namespace abbreviation) that shall be used to represent the annotation.

8.2 Annotation Encodings for the Ontology

Table 4 – Annotation Properties for Metadata about the Ontology

Ref.	Type of Documentation	Multiplicity (M/C/O)	Value Type	Notes	Annotation Property (*namespace*:*property*)
1	Ontology IRI	Exactly one (Mandatory)	IRI	The unique character sequence used as the identifier of the ontology (owl:Ontology). NOTE: The ontologyIRI followed by the separator "#" denotes the ontology namespace.	owl:ontologyIRI
2	Title	Exactly one (Mandatory)	Localized character string	The preferred human-readable designation of a resource.	dct:title [Alternatively: skos:prefLabel]
3	Comment	Exactly one (Mandatory)	Localized character string	The purpose and scope of the resource.	rdfs:comment
4	Version IRI	If applicable, exactly one	IRI	The unique character sequence used to identify a particular version of the ontology.	owl:versionIRI
5	Version Information	Exactly one (Mandatory)	Character string	The ontology version number in the structured-string format specified for ontology management.	owl:versionInfo
6	Prior Version	Zero or more (Optional)	IRI	A prior version (specified by IRI) of the annotated ontology.	owl:priorVersion
7	Backward Compatible With	Zero or more (Optional)	IRI	A prior version (specified by IRI) of the annotated ontology that is compatible with the current version of the ontology.	owl:backwardCompatible With
8	Incompatible With	Zero or more (Optional)	IRI	A prior version (specified by IRI) of the annotated ontology that is not compatible with the current version of the ontology.	owl:incompatibleWith

8.3 Annotation Encodings for Ontology Components

Table 5 – Annotation Properties for Metadata about Ontology Components

Ref.	Type of Documentation	Multiplicity (M/C/O)	Value Type	Notes	Annotation Property (*namespace*:*property*)
1	Concept IRI	Exactly one (Mandatory)	IRI	The IRI of the ontology component (OWL Class or Property). NOTE: Enumerant literals do not have *IRIs*.	rdf:about
2	Label	Exactly one (Mandatory)	Localized continuous string	The camel-cased string equivalent to the terminal segment of the concept IRI. NOTE: Enumerant literals do not have *labels*.	rdfs:label
3	Name	Exactly one (Mandatory)	Localized character string	The human-readable designation for a concept. EXCEPTION: The name for an Enumerant literal is a string in a list structure and may be without the language indicator.	skos:prefLabel
4	Alternate Name or Abbreviation (Alias)	Zero or more (Optional)	Localized character string	An alternate name, acronym, or abbreviation for a concept. NOTE: Enumerant literals do not have *aliases*.	skos:altLabel
5	Definition	Exactly one (Mandatory)	Localized character string	A precise, succinct statement of the meaning of a concept specifying its essential characteristics. NOTE: Enumerant literals do not have *definitions*.	skos:definition
6	Description	Zero or more (Optional)	Localized character string	Additional information about the meaning of a concept, beyond the definition; for example: non-essential characteristics, variations, scope, and additional context. NOTE: Enumerant literals do not have *descriptions*.	dct:description
7	Example	Zero or more (Optional)	rdfs:Resource (String or IRI)	Illustration or description of the use of a concept. NOTE: Enumerant literals do not have *examples*.	skos:example

9 Conclusions

9.1 Importance of Annotations

Human-understandable annotations are used in the CASE ontology to support users in learning about and applying the ontology. Human-readable names and definitions for ontology concepts assist in standardizing a vocabulary for cyber objects and cyber-investigation activities. Names encoded in annotations may be accessed for use as display labels in user interfaces by programs able to process the annotations. Human-understandable definitions support the interpretation of CASE-compliant data and the development of mappings and services useful to adopters of the ontology.

9.2 Flexibility of Annotations

The set of annotations specified in this guidance document are intended to support the basic requirements for designating and understanding concepts in the CASE ontology. The availability of several metadata standards compatible with RDF resources allows for additional annotation properties to be added to CASE, as needed, through the normal change-request processes. Examples from some of those standards may be seen in Annex B.

9.3 Publication of Annotations

CASE concept designators (names) and definitions contained in annotations may be used to generate HTML documentation for the ontology, by using a tool such as OWLDoc in Protégé. In this way, the annotations may be used to publish a controlled vocabulary for CASE. A controlled vocabulary is a set of terms consisting of defined lexical items (*i.e.*, words, phrases, or abbreviations from a natural language) that are collected and managed by an authority following identified criteria for inclusion. Controlled vocabularies promote consistent semantics in applications that provide terminology services, tagging and/or indexing, search, display of resources, and related Web services for data sharing. They also support human understanding of data models and structured information – including the CASE ontology and CASE-compliant data.

Annex A: Ontology Component Categories
(Informative)

The vocabulary information model defines categories for ontology components according to the type of their denotation, that is, the kind of thing they represent in the domain model. These categories are different from the constructs of the OWL ontology language but are correlated to them.

Table 6 – Categories Used to Describe Ontology Components

Ref #	Name	Definition	OWL Construct
1	Entity Class	The ontology component represents a type of real-world phenomenon. NOTE: An entity class is an abstraction that characterizes a set of individuals. EXAMPLES: “Investigation”, “Person”, “Tool”, “Event”, “Marriage”.	owl:Class
2	Property	The ontology component represents either: (a) a characteristic of an entity, or (b) a relationship between two entities (excluding relationships that are represented as entity classes). NOTE 1: A characteristic will have a specific value in relation to a particular entity at a particular time. EXAMPLES 1: (1) "Length" (for a physical object, with a value in physical distance); (2) “Color” (for a physical object, with a specified value either in a qualitative color range, or in a wavelength of light); (3) “Principal Activity” (for an organization, with an activity type as value). NOTE 2: A relationship will hold between two entities at a particular time. EXAMPLES 2: (1) A relationship of one agent (*i.e.*, a person or organization) to another agent, in which the first agent (*employer*) provides financial compensation to the second agent in return for an assigned task or role. The inverse relationship would be *employee*. (2) A relationship of a person to a country, whereby the person is a *citizen-of* the country.	owl:DatatypeProperty or owl:ObjectProperty
3	Enumeration	The ontology component represents a value domain as a set of allowablevalues for a property, in which all possible values (*i.e.*, enumerants) are listed. An enumeration may be a set of individuals or a set of literals. EXAMPLE: The enumeration “Byte Order”, comprising the values “Big Endian” and “Little Endian”.	owl:Class
4	Enumerant	The ontology component represents a value from an enumeration. NOTE: Enumerants may be represented either by Named Individuals (in an Enumeration of Individuals) or by literals (in an Enumeration of Literals).[17] EXAMPLE (Possible values for characterizing a problem report about a software program): “Critical”, “Debug”, “Error”, “Info”, or “Warning”.	Literal (*e.g.*, xsd:string) or owl:NamedIndividual (depending upon the type of Enumeration)

[17] OWL 2 Structural Specification (11 December 2012), Section 7.4; OWL 2 Mapping to RDF (11 December 2012), Table 1.

Annex B: Annotation Properties

(Informative)

Table 7 – Annotation Properties from (non-OWL) W3C and Other Standards

Ref	Standard	Annotation Property	Usage	Domain	Range
1	RDFS 1.1[18] (3.6)	rdfs:label	Used to provide a human-readable version of a resource's name.	rdfs:Resource	rdfs:Literal
2	RDFS 1.1 (3.7)	rdfs:comment	Used to provide a human-readable description of a resource.	rdfs:Resource	rdfs:Literal
3	SKOS Ref (2009)[19]	skos:note	Used for general documentation purposes.	[rdfs:Resource][20]	[rdfs:Resource]
4	SKOS Ref (2009)	skos:changeNote	Used to document fine-grained changes to a concept.	[rdfs:Resource]	[rdfs:Resource]
5	SKOS Ref (2009)	skos:definition	Used to provide a complete explanation of the intended meaning of a concept.	[rdfs:Resource]	[rdfs:Resource]
6	SKOS Ref (2009)	skos:editorialNote	Used to provide information about administrative housekeeping, such as a reminder of editorial work to be done.	[rdfs:Resource]	[rdfs:Resource]
7	SKOS Ref (2009)	skos:example	Used to provide an example of the use of a concept.	[rdfs:Resource]	[rdfs:Resource]
8	SKOS Ref (2009)	skos:historyNote	Used to describe significant changes to the meaning or the form of a concept.	[rdfs:Resource]	[rdfs:Resource]
9	SKOS Ref (2009)	skos:scopeNote	Used to provide some, possibly partial, information about the intended meaning of a concept esp. in applications.	[rdfs:Resource]	[rdfs:Resource]

[18] RDF Schema 1.1 (25 February 2014) defines the content of the "rdfs" namespace. See Sections 3.6 (rdfs:label) and 3.7 (rdfs:comment). Also see Sections 2.1 (rdfs:Resource) and 2.3 (rdfs:Literal) for the domain and range. Available online at https://www.w3.org/TR/rdf-schema/. Note that rdf:PlainLiteral (http://www.w3.org/TR/rdf-plain-literal/) was the datatype proposed and standardized by the W3C OWL community before the RDF Community created RDF-1.1 (https://github.com/rdfjs/N3.js/issues/15).

[19] SKOS Simple Knowledge Organization System Reference (18 August 2009) defines the content of the "skos" namespace. Available online at https://www.w3.org/TR/skos-reference/. Sections 5 (Lexical Labels) and 7 (Documentation Properties) define the SKOS annotation properties. Also see the SKOS Primer (https://www.w3.org/TR/skos-primer/), Section 2.4 (Documentary Notes).

[20] No domain or range is stated for SKOS Notes, therefore making the effective domain and range for these properties the class of all resources (*i.e.*, rdfs:Resource). SKOS Reference, Section 7.5.

10	Dublin Core Terms[21]	dct:accessRights	Used to document information about who can access a resource or an indication of its security status.	[rdfs:Resource]	dct:RightsStatement
11	DC Terms	dct:description	Used to provide a (typically free-text) account of a resource.	[rdfs:Resource]	rdfs:Literal
12	DC Terms	dct:license	Used to relate to a legal document giving official permission to do something with the resource.	[rdfs:Resource]	dct:LicenseDocument
13	DC Terms	dct:source	Used to link to a resource from which the annotated resource is derived.	[rdfs:Resource]	[IRI]
14	DC Terms	dct:title	Used to provide the name given to the resource.	[rdfs:Resource]	rdfs:Literal

[21] Dublin Core Metadata Initiative (DCMI) Terms defines the content of the "dct" (DC Terms) namespace. Available online at: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/2012-06-14/ (accessed 8/1/2019).