NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY

NGA.STND.0065_1.0_NTAX
2017-04-11

NGA STANDARDIZATION DOCUMENT

National System for Geospatial Intelligence Taxonomy (NTAX) Standard
(2017-04-11)

Edition 1.0

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

NATIONAL CENTER FOR GEOSPATIAL INTELLIGENCE STANDARDS

Table of Contents

Introduction ..... **iv**
1 Scope **1**
2 Conformance **1**
2.1 Conformance Requirements 1
2.2 Abstract Test Suite 1
3 References **1**
3.1 Normative 1
3.2 Informative 2
4 Terms, Definitions, and Acronyms **3**
4.1 Terms and Definitions 3
4.2 Acronyms 6
4.3 Presentation Font 6
5 Taxonomy Specification **7**
5.1 Introduction 7
5.2 NSG Taxonomy Information Model 7
5.2.1 Introduction 7
5.2.2 Relationship to the ISO 19109 General Feature Model 7
5.2.3 NTAX Information Model Concepts and Definitions 8
5.2.4 Datatypes in the NSG Taxonomy Information Model 13
5.3 NSG Taxonomy Representation using Semantic Web Languages 14
5.3.1 The Semantic Web 14
5.3.2 Selecting OWL Constructs for Representation of NTAX Content 15
5.3.3 Representing NTAX Information Model Concepts in OWL 16
5.3.4 Unique Identifiers in OWL: IRIs 16
5.3.5 Structural Elements of the NSG Taxonomy 17
5.3.6 Documentation Properties in the NSG Taxonomy 17
5.4 NSG Taxonomy Encodings 18
5.4.1 Introduction 18
5.4.2 Namespace and Identifiers 18
5.4.3 General NTAX Encoding 20
5.4.4 General Encoding of Datatypes 23
5.4.5 Technology-specific NTAX Encodings 24
6 Governance **28**
6.1 Introduction 28
6.2 Official Publication of the NTAX Standard and NTAX Content 28
6.2.1 Introduction 28
6.2.2 Publication of NTAX Content as a Technical Artifact 28
6.2.3 Publication of NTAX Content as REST API-accessible Resources 28
Annex A – Conformance (Normative) **31**
A.1 Introduction 31
A.1.1 Terms and Definitions 31
A.1.2 Conformance Testing Methodology 32
A.1.3 Testing Conformance for a Subset of the NSG Taxonomy 33
A.1.4 Testing Conformance for the NSG Taxonomy with Extensions 34
A.1.5 Logical Structure of the Abstract Test Suite for the NSG Taxonomy 34
A.2 Abstract Test Suite for the NSG Taxonomy (NTAX) Standard 36
A.2.1 Test Module for Conformance with Taxonomy Structure 36
A.2.1.1 Test Case for Taxonomy Dependency(ies) 36
A.2.1.2 Test Module for Components with Identity (IRIs) 36
A.2.1.2.1 Test Case for Taxonomy with IRI 36
A.2.1.2.2 Test Case for Entity Classes with IRIs 37
A.2.1.2.3 Test Case for DisjointClasses with IRIs 37

A.2.1.3 Test Module for Generalization (Subclass) Hierarchy ... 37
A.2.1.3.1 Test Case for Top Entity Classes ... 37
A.2.1.3.2 Test Case for Generalization Relationships ... 38
A.2.1.4 Test Module for Disjointness Axioms ... 38
A.2.1.4.1 Test Module for Disjoint Collection or List ... 38
A.2.1.4.1.1 Test Case for DisjointClasses Collection ... 38
A.2.1.4.1.2 Test Case for DisjointClasses List ... 39
A.2.1.4.2 Test Case for Members of DisjointClasses ... 39
A.2.1.4.3 Test Case for Skolemized IRIs ... 39
A.2.2 Test Module for Documentation of Semantics ... 39
A.2.2.1 Test Module for Taxonomy Documentation ... 40
A.2.2.1.1 Test Case for Taxonomy Version Information ... 40
A.2.2.1.2 Test Case for Taxonomy Label ... 40
A.2.2.1.3 Test Case for Taxonomy Name ... 40
A.2.2.1.4 Test Case for Taxonomy Alias ... 40
A.2.2.1.5 Test Case for Taxonomy Definition Note ... 41
A.2.2.1.6 Test Case for Taxonomy Source Reference ... 41
A.2.2.1.7 Test Case for Taxonomy Source Title ... 41
A.2.2.2 Test Module for EntityClass Documentation ... 41
A.2.2.2.1 Test Case for Abstract Class ... 41
A.2.2.2.2 Test Case for EntityClass Label ... 42
A.2.2.2.3 Test Case for EntityClass Name ... 42
A.2.2.2.4 Test Case for EntityClass Alias ... 42
A.2.2.2.5 Test Case for EntityClass Definition Note ... 42
A.2.2.2.6 Test Case for Source Reference ... 42
A.2.2.2.7 Test Case for Part-of ... 43
A.2.3 Test Module for Datatype Conformance ... 43
A.2.3.1 Test Case for IRI Datatype ... 43
A.2.3.2 Test Case for Boolean Datatype ... 43
A.2.3.3 Test Case for CharacterString Datatype ... 43
A.2.3.4 Test Case for LocalizedCharacterString Datatype ... 44
A.2.3.5 Test Case for LocalizedContinuousString Datatype ... 44
A.2.3.6 Test Case for IANALanguageSubtag Datatype ... 44
A.2.4 Test Module for Categorization ... 44
A.2.4.1 Test Case for Declarations of Instances ... 45
A.2.4.2 Test Case for Semantics of Instances ... 45

Annex B – Inspecting NSG Taxonomy (NTAX) Content (Informative) ... 46
B.1 Introduction ... 46
B.2 NTAX Content Inspection using the Protégé Ontology Tool ... 46
B.2.1 Protégé: An Open-Source Ontology Tool for Viewing W3C OWL ontologies ... 46
B.2.2 Viewing NTAX Content using Protégé and its Plug-ins ... 46

Annex C – Derivation of the NSG Taxonomy (Informative) ... 51
C.1 Introduction ... 51
C.2 NTAX Structural Requirements ... 51
C.3 NTAX Encoding Element Derivation ... 52

Annex D – UML Primer (Informative) ... 55
D.1 UML Notations ... 55
D.2 UML Model Relationships ... 55
D.2.1 Associations ... 55
D.2.2 Navigation ... 55
D.2.3 Generalization ... 55
D.2.4 Instantiation / Dependency ... 56
D.2.5 Roles ... 56
D.3 UML Model Stereotypes ... 56

Table of Figures

Figure 1 – Entity Generalization in the ISO 19109 General Feature Model 8
Figure 2 – Overview of the NTAX Information Model 9
Figure 3 – The Semantic Web Stack....15
Figure 4 – OWL2 RDF/XML Encoding: Entity Class `Aerodrome` and its Disjoint Subclasses....24
Figure 5 – OWL 2 RDF/XML Encoding: Subclasses of `Aerodrome`....25
Figure 6 – N-Triples Encoding: Entity Class `LandAerodrome`....27
Figure 7 – N-Triples Encoding: Disjoint Subclasses of `Aerodrome`....27
Figure 8 – Sample of NTAX individual resource (Entity Class “Hulk”) in N-Triples encoding....30
Figure 9 – Structure of Abstract Test Suite for NSG Taxonomy Standard 35
Figure 10 – NTAX Described on the Active Ontology Tab of the Protégé Tool....47
Figure 11 – Protégé View of NTAX Hierarchy with Individual Concept (Feature Entity)....48
Figure 12 – OntoGraf Plug-in View of NTAX Actor Entity Hierarchy with Documentation....49
Figure 13 – OWLViz Plug-in View of Subclasses of NTAX Building....50
Figure 14 – Top Entity Classes of the NSG Taxonomy....52
Figure 15 – UML Notation 55
Figure 16 – UML Roles 56

Table of Tables

Table 1 – Normative References....2
Table 2 – Informative References 2
Table 3 – Definitions Applicable to this Standard....3
Table 4 – Definitions of Modeling Concepts in the NTAX Information Model 11
Table 5 – NTAX Encoding Elements for the Taxonomy 21
Table 6 – NTAX Encoding Elements for an Entity Class....22
Table 7 – NTAX Encoding Elements for Disjoint Classes 23
Table 8 – Terms and Definitions for Conformance Testing 31
Table 9 – Derivation of NTAX Encoding Elements for an Entity Class....53
Table 10 – Derivation of NTAX Encoding Elements for Disjoint Classes 54

Introduction

The NSG Taxonomy (NTAX) Standard document ("NTAX Standard") defines the specification for a hierarchy of entity classes to represent concepts used for the typing of Geospatial Intelligence (GEOINT) information shared in the U.S. National System for Geospatial Intelligence (NSG). This hierarchy is a taxonomy – that is, a tree-like structure for classification – represented using subclass relationships as defined in the *World Wide Web Consortium (W3C) Web Ontology Language, Second Edition* (OWL 2). This NTAX Standard specifies the information model, two encoding patterns, and a governance process for managing the taxonomy.

The content of the NSG Taxonomy ("NTAX content") consists of specific entity classes to be used for data classification. The NTAX content is derived from a subset of the content of the logical model for geospatial data in the NSG enterprise, the NSG Application Schema (NAS). While the NAS is a complete application schema for defining NSG-specific data sets, the NSG Taxonomy uses only the NAS content that defines entity types (for example: features, actors, events). The NTAX content is specified separately from the NTAX Standard in officially published technical artifacts. The technical artifacts are implemented in two of the W3C encodings defined for OWL 2 – Resource Description Format (RDF) XML and N-Triples.

The NTAX enables the semantics (*i.e.*, meaning) of GEOINT data published on the Web to be represented based on well-known International Standards and W3C Recommendations.[1] In this way, the NTAX supports collaborative efforts across the U.S. National System for Geospatial Intelligence (NSG) to build a linked open data store of GEOINT data with integrated, machine-processable semantics. The specific purpose of the NTAX content is for the categorization of resources (*e.g.*, documents, objects, and data instances) using a common semantic framework, which supports improved management, search, retrieval, and utilization of those resources. The NTAX promotes data interoperability between applications that enable categorization, indexing, search, navigation, content presentation, constraint assertion, and Web services.

Both the NTAX Standard and NTAX content are developed and managed under the authority of the GEOINT Content Standards Board (GCSB) of the National Geospatial-Intelligence Agency (NGA). The NTAX Standard and the two NTAX content encodings are published as registered resources in the online NSG-unique Standards Register of the NSG Standards Registry. The encoded NTAX content is also available as individual concept resources retrievable through the REST API component of the NSG Standards Registry.

Revision History

Description	Date	Edition
Initial Edition	04/11/2017	1.0
Minor editorial corrections in Sections 4.1 (Table 3); 5.3.6; 5.4.2.3; 5.4.3.3 (Table 5); and 6.2.2. Section 5.3.6, first paragraph, revised for clarity.	09/01/2017	1.0

[1] *Data on the Web Best Practices, a W3C Recommendation* (31 January 2017). Latest version available online at: http://www.w3.org/TR/dwbp/. Guidance on the production of data instances falls outside the scope of this standard.

1 Scope

The NSG Taxonomy (NTAX) Standard document ("NTAX Standard") defines the specification for a hierarchy of entity classes to represent concepts used for the typing of Geospatial Intelligence (GEOINT) information shared in the U.S. National System for Geospatial Intelligence (NSG).[2] This hierarchy is a taxonomy – that is, a tree-like structure for classification – represented using subclass relationships as defined in the *World Wide Web Consortium (W3C) Web Ontology Language, Second Edition* (OWL 2). The NTAX enables the semantics (*i.e.*, meaning) of GEOINT data published on the Web to be represented based on well-known International Standards and W3C Recommendations. This NTAX Standard specifies the information model for the taxonomy, along with two encoding patterns; these are presented in Section 5.

The content of the NSG Taxonomy ("NTAX content") consists of specific entity classes for the classification of resources (*e.g.*, documents, objects, and data instances) using a common semantic framework, which supports improved management, search, retrieval, and utilization of those resources. The NTAX content is specified separately from the NTAX Standard in officially published technical artifacts. The technical artifacts are implemented in two of the W3C encodings defined for OWL 2 – Resource Description Format (RDF) XML and N-Triples.[3]

The NTAX Standard is presented in support of collaborative efforts across the NSG to build a linked open data store of GEOINT data with machine-processable semantics. The NTAX content promotes data interoperability between applications that enable categorization, indexing, search, navigation, content presentation, constraint assertion, and Web services for GEOINT data.

Guidance on the production of data instances falls outside the scope of this standard.

Both the NTAX Standard and NTAX content are developed and managed under the authority of the GEOINT Content Standards Board (GCSB) of the National Geospatial-Intelligence Agency (NGA). The NTAX Standard and NTAX content evolve in response to NSG community requirements. The NTAX Standard and the two NTAX content encodings are published as registered resources in the online NSG-unique Standards Register of the NSG Standards Registry hosted by the National Geospatial-Intelligence Agency (NGA). The NSG Standards Registry is the single authoritative source for the NTAX Standard and for the technical artifacts encoding the NTAX content in RDF/XML and N-Triples format. The encoded NTAX content is also available as individual concept resources retrievable through the REST API component of the NSG Standards Registry. The governance process is specified in Section 6.

The NGA is the authority for promulgating the NTAX Standard and its accompanying technical artifacts encoding the NTAX content for use by the U.S. Department of Defense (DoD), U.S. Intelligence Community (IC), and U.S. civil federal agencies.

2 Conformance

2.1 Conformance Requirements

Any product claiming conformance with this NTAX Standard and/or the accompanying NTAX content shall pass all of the requirements stated in the abstract test suite (ATS) in Annex A, which enumerates the specific elements of conformance.

2.2 Abstract Test Suite

The abstract test suite (ATS) for the NTAX is specified in Annex A (normative). Annex A also describes conditions under which subsets of the NTAX content may be used, and for using the NTAX content with externally defined extensions.

3 References

3.1 Normative

The documents listed in Table 1 are indispensable to understanding and using this standard. 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.

[2] This edition of the NTAX Standard is available from the NSG Standards Registry, at: http://nsgreg.nga.mil/doc/view?i=2616.
[3] RDF/XML is the mandated encoding for OWL 2. N-Triples is an optional, plain-text format for encoding an RDF graph.

Table 1 – Normative References

Standard or Specification
NSG Taxonomy (NTAX) content, encoded in technical artifacts: http://nsgreg.nga.mil/ntax
ISO 19150-2:2015. Geographic information – Ontology – Part 2: Rules for developing ontologies in the Web Ontology Language (OWL): https://www.iso.org/standard/57466.html
IETF RFC 3987, *Internationalized Resource Identifiers (IRIs)*: http://www.ietf.org/rfc/rfc3987.txt
IETF RFC 4646, BCP 47, *Tags for Identifying Languages*: http://www.ietf.org/rfc/bcp/bcp47.txt
W3C *OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition)*, 11 December 2012: http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/
W3C *OWL 2 Web Ontology Language: Mapping to RDF Graphs (Second Edition)*, 11 December 2012: http://www.w3.org/TR/2012/REC-owl2-mapping-to-rdf-20121211/
W3C *RDF 1.1 Concepts and Abstract Syntax*, 25 February 2014: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/
W3C *RDF 1.1 N-Triples: A line-based syntax for an RDF graph*, 25 February 2014: http://www.w3.org/TR/n-triples/
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/
W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes (5 April 2012): http://www.w3.org/TR/xmlschema11-2/
DCMI Metadata Terms (14 June 2012): http://dublincore.org/documents/dcmi-terms/2
GEOINT Content Standards Board (GCSB) Operations Guide. NGA.SIG.0029_1.0_GCSB: http://nsgreg.nga.mil/doc/view?i=4284

3.2 Informative

The informative (non-normative) documents listed in Table 2 are useful to understanding and using this standard. For dated references, only the cited edition or version applies.

Table 2 – Informative References

Standard or Specification
ISO 19101-1:2014. *Geographic information – Reference model – Part 1: Fundamentals* (November 2014): https://www.iso.org/standard/59164.html
ISO 19105:2000. *Geographic information – Conformance and testing* (December 2000): https://www.iso.org/standard/26010.html
ISO 19109:2015. *Geographic information – Rules for application schema* (December 2015): https://www.iso.org/standard/59193.html
ISO 19110:2016. *Geographic information – Methodology for feature cataloguing* (December 2016): https://www.iso.org/standard/57303.html
ISO/IEC 10646:2012, *Information technology – Universal Coded Character Set (UCS)*: https://www.iso.org/standard/56921.html

Standard or Specification
IETF RFC 1738, *Uniform Resource Locators (URL)*: http://www.ietf.org/rfc/rfc1738.txt
IETF RFC 3986, *Uniform Resource Identifiers (URI): Generic Syntax*: http://www.ietf.org/rfc/rfc3986.txt
OGC Testbed-12 ShapeChange Engineering Report. 16-020. November 2016.
OMG Unified Modeling Language (OMG UML), Infrastructure, Version 2.2: http://www.omg.org/spec/UML/2.2/Infrastructure/PDF/
OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.2: http://www.omg.org/spec/UML/2.2/Superstructure/PDF/
National System for Geospatial Intelligence Application Schema (NAS) – Part 1: Platform Independent Model: http://nsgreg.nga.mil/nas
Shorter Oxford English Dictionary, *Sixth Edition* (version 3.0.2.1). CD-ROM.

4 Terms, Definitions, and Acronyms

4.1 Terms and Definitions

The terms and definitions[4] specific to this standard are presented in Table 3.

Table 3 – Definitions Applicable to this Standard

Term	Definition
annotation	An expression used to associate information with an **ontology**. NOTE1: An annotation is additional information associated to ontologies or ontology components that is intended for human consumption and not for use by reasoning software. NOTE2: Each annotation consists of an **annotation property** and an **annotation value**. SOURCE: OWL 2 Structural Specification (Section 3.5)
annotation property	A model element used to provide a textual **annotation** for an **ontology** or ontology component. SOURCE: OWL 2 Structural Specification (Section 5.5).
annotation value	A literal (including character strings), an **IRI**, or an **anonymous individual** that is the value of an **annotation property**. SOURCE: OWL 2 Structural Specification (Section 3.5)
anonymous individual	In OWL 2, an individual that is identified by a local node ID rather than a global **IRI**, and which is not expected to be used or referred to outside of a particular **ontology**. NOTE1: Individuals in OWL 2 represent objects from the domain. SOURCE: OWL Structural Specification (Section 5.6)
axiom	A statement of something that is true in the **universe of discourse** (domain). NOTE: 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: OWL 2 Structural Specification (Section 9)
blank node	A node in an RDF graph that is distinct but has no **IRI** identifier. NOTE: A blank node cannot be referred to outside of its local graph. When stronger identification is needed, a blank node may be replaced and represented in the graph with a new, globally unique IRI (a Skolemized IRI) corresponding to the blank node. SOURCE: RDF 1.1 Concepts and Abstract Syntax (Sections 3.4, 3.5)

[4] In definitions, a term is styled in **bold** when the meaning of that term is specified elsewhere in Section 4.1.

Term	Definition
change notification (regarding a standard)	A publication in which modifications to selected items in a standard are reported in detail to the community of its users by the applicable maintenance authority. NOTE: In the management of the NTAX content, a change notification establishes a new **content baseline**. SOURCE: GCSB Operations Manual
class [OWL]	A set of individuals. SOURCE: OWL 2 Structural Specification (Section 5.1)
class [UML]	A description of a set of objects that share the same attributes, operations, methods, relationships, and semantics. NOTE: A class represents a concept being modelled. SOURCE: Based on ISO 19103:2005 (Clause G.3)
class expression	A logic-based description composed from one or more **class**es and property expressions that represents a set of individuals (*i.e.*, a **class**) by formally specifying the condition(s) on the properties of individuals belonging to the class. NOTE1: Individuals that satisfy the specified conditions are said to be instances of the class expression. SOURCE: OWL Structural Specification (Section 8)
conceptual model	A model that defines concepts of a **universe of discourse**. SOURCE: ISO 19101-1:2014 (Clause 4.1.5)
content baseline (of a standard)	The complete set of content of a standard, which is authorized (*i.e.*, valid) for use at a specified time. NOTE1: A content baseline is established by publication of a technical artifact containing the content that is valid at that time. NOTE2: A content baselines may be established concurrent with the publication of a new edition of a standard, or solely based on changes to the content of a standard. SOURCE: GCSB Operations Manual (Section 2.3.5)
edition (of a standard)	A publication containing the entire current content of an established standard, and issued by the authorized publication authority, either as the first edition of a new standard or as a new edition (*i.e.*, revised complete version, usually numbered; for example, "2nd edition") of a previously published standard. SOURCE: GCSB Operations Manual
entity class	A modeling class that represents a **feature** or other geospatially-located information. SOURCE: Based on *entity*, NAS – Part 1 (Section 1.1)
feature	An abstraction of real-world phenomena. NOTE1: ISO 19101, *Geographic information – Reference Model,* defines a feature as an abstraction of real world phenomena. Such abstractions may be represented in information systems using a variety of spatial modeling methods, including representations such as vectors, grids and images. SOURCE: ISO/TC211 19101:2014 (Clause 4.1.11) NOTE2: The NSG Application Schema (NAS) – Part 1 also supports modeling entities that may represent other geospatially-located information that does not correspond to “real world phenomena”.
generalization [UML]	A taxonomic relationship between a more general element and a more specific element of the same element type. NOTE: An instance of a more specific element may be used where its more general element is allowed. SOURCE: ISO/TS 19103:2005 (Clause 4.2.13) (19103:2015, Clause 4.18) [based on UML 2]

Term	Definition
inheritance [UML]	The mechanism by which more specific classifiers incorporate structure and behavior defined by more general classifiers. SOURCE: ISO/TS 19103:2005 (Clause 4.19)
Internationalized Resource Identifier (IRI)	A sequence of characters from the Universal Character Set (Unicode/ISO 10646) [IETF RFC 3987], intended for identifying an abstract or physical resource. NOTE1: 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: IETF RFC 3987 NOTE2: A resource can be anything that has identity, *e.g.*, an OWL class instance and its associated annotations. NOTE3: OWL 2 extends OWL 1 by using IRIs to identify ontologies and their elements. OWL 1 uses Uniform Resource Identifiers (URI). SOURCE (NOTE3): OWL 2 Structural Specification (Section 2.4)
namespace	In RDF, a common **URI** prefix or stem used in identifiers for a set of related resources. NOTE1: The RDF namespace is concatenated with the local name to create the complete URI identifier for an RDF resource. NOTE2: Every RDF resource is identified by a URI. SOURCE: ISO 19150-2:2015 NOTE3: The NTAX Standard and NTAX content encodings use the standard prefix names for namespaces as declared in the OWL Structural Specification (Section 2.4).
ontology	A formal representation of phenomena of a universe of discourse with an underlying vocabulary including definitions and **axiom**s that make the intended meaning explicit and describe phenomena and their interrelationships. EXAMPLES: Basic Formal Ontology (BFO); Suggested Upper Merged Ontology (SUMO); Friend of a Friend (FOAF). NOTE: An ontology represents a universe of discourse (whether the world-at-large or a specific domain) with a formalization based in logical theory, including formally defined concepts and properties, with restrictions and optionally rules, in which the structural relationships (including *subclass-of*, *equivalence*, and *disjoint-with*) are defined in a formal logic (either axiomatically or in a rule-based formulation). SOURCE: ISO 19150-2 citing ISO 19101-1:2014, 4.1.26
subclass	A relationship between two **class**es such that each instance of the more specific class is also an instance of the more general class. NOTE1: In OWL, an axiom expressed with “SubClassOf” states that each instance of one class is also an instance of another, more general class. NOTE2: Subclass axioms may be used to construct a hierarchy of classes. SOURCE: OWL Structural Specification (Section 9.1)
taxonomy	The classification of information entities in the form of a hierarchy, according to the presumed relationships of the real-world entities that they represent. NOTE: A taxonomy of **class**es arranged in a hierarchy using the **subclass** relationship to represent **generalization** is a strict (or “strong”) taxonomy, in which **inheritance** applies. EXAMPLE: Linnaean classification (strong taxonomy). SOURCE: *The Semantic Web*. Michael C. Daconta, Leo J. Obrst, and Kevin T. Smith. Wiley Publishing, Inc. 2003.
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).

Term	Definition
Uniform Resource Identifier (URI)	A compact string of characters for identifying an abstract or physical resource. NOTE1: A resource can be anything that has identity, *e.g.*, an OWL class instance and its associated annotations. NOTE2: A URI identifies a resource either by location, or by name, or both. NOTE3: 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 RFC1738
universe of discourse	A view of the real or hypothetical world that includes everything of interest. SOURCE: ISO 19150-2 citing ISO 19101-1:2014 (Clause 4.1.38)

4.2 Acronyms

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

API Application Programming Interface
BCP Best Current Practice
DCMI Dublin Core Metadata Initiative
DoD (U.S.) Department of Defense
GCSB GEOINT Content Standards Board
GEOINT Geospatial Intelligence
GFM General Feature Model
IANA Internet Assigned Numbers Authority
IC (U.S.) Intelligence Community
IEC International Electrotechnical Commission
IRI Internationalized Resource Identifier
ISO International Organization for Standardization
IETF Internet Engineering Task Force
NAS NSG Application Schema
NGA National Geospatial-Intelligence Agency
NSG National System for Geospatial Intelligence
NTAX NSG Taxonomy
OGC Open Geospatial Consortium
OMG Object Management Group
OWL Web Ontology Language
RDF Resource Description Language
RDFS RDF Schema
REST REpresentational State Transfer
SKOS Simple Knowledge Organization System
UML Unified Modeling Language
URI Uniform Resource Identifier
URL Uniform Resource Locator
W3C World Wide Web Consortium
XML Extensible Markup Language

4.3 Presentation Font

The general text of this document is presented using Arial font. Encoding elements for the NSG Taxonomy are presented using Courier New font (*e.g.*, `owl:Class`).

5 Taxonomy Specification

5.1 Introduction

The NSG Taxonomy (NTAX) Standard establishes the terminological, semantic, and structural basis for specifying a generalization hierarchy of entity concepts that may be used for the categorization of geospatial data in the National System for Geospatial Intelligence (NSG). This section on taxonomy specification defines the information model for representing the content of the NSG Taxonomy and specifies two encodings for that content.

The NTAX Standard specifies an information model based on the General Feature Model (GFM) of ISO 19109:2015. The formal representation for the NSG Taxonomy is based on the family of Web Ontology Language, Second Edition (OWL 2) Recommendations (*i.e.*, standards) of the World Wide Web Consortium (W3C). Representing entity concepts as OWL 2 logical classes supports the use of concepts in the NTAX content to describe data exchanged among automated information systems in a machine-processable way.[5]

To support usage on the Web, the NTAX Standard prescribes Internationalized Resource Identifiers (IRIs) to identify the NSG Taxonomy and its components. To support applications that require a Web encoding format, the NTAX Standard specifies two W3C encodings for the NSG Taxonomy. The encoded NTAX content is published in the NSG-unique Standards Register of the NSG Standards Registry and is also accessible through the REST API component of the NSG Standards Registry.

The taxonomy specification in the NTAX Standard defines the NSG Taxonomy in three ways, by providing:

1. Content specification – the information model for NSG Taxonomy entity classes (types) and their generalization relationships;
2. Content identification – IRIs for unique identification of the NSG Taxonomy as a whole and for its components (NTAX content); and
3. Content encoding – Specifications for the RDF/XML and N-Triples encodings of the NTAX content.

Section 5.2 specifies the information model for the NSG Taxonomy in a diagram together with a tabular specification of all included modeling elements.

Section 5.3 specifies the use of the W3C representation language OWL 2 to represent elements of the information model.

Section 5.4 specifies the implementation of the NTAX content in two supported encodings: (1) RDF/XML; and (2) N-Triples. This allows instance data in either RDF/XML or N-Triples formats to be linked to concepts in the NSG Taxonomy to provide semantics for data exchanged among automated information systems.

5.2 NSG Taxonomy Information Model

5.2.1 Introduction

The information model for the NSG Taxonomy defines the modeling concepts needed to represent a generalization hierarchy of entity types used for the categorization of geospatial data in the National System for Geospatial Intelligence (NSG). The NTAX information model is conceptually related to the sub-part of the General Feature Model (ISO 19109) containing only those modeling elements needed to represent a generalization hierarchy of entity concepts.

Section 5.2.2 specifies those portions of the General Feature Model (ISO 19109) that are used in the NTAX information model. Non-hierarchical relationships and entity properties in the General Feature Model are not included in the NTAX information model.

The NTAX information model also specifies a set of documentation properties to record the semantics and provenance of entity classes.

Section 5.2.3 specifies the modeling elements of the NTAX information model in the form of a diagram (Figure 2) and a tabular specification of all included modeling concepts.

5.2.2 Relationship to the ISO 19109 General Feature Model

The NSG Taxonomy information model is based on a core fragment of the General Feature Model (GFM) defined in ISO 19109.

[5] *Data on the Web Best Practices, a W3C Recommendation* (31 January 2017). Available online at: http://www.w3.org/TR/dwbp/.

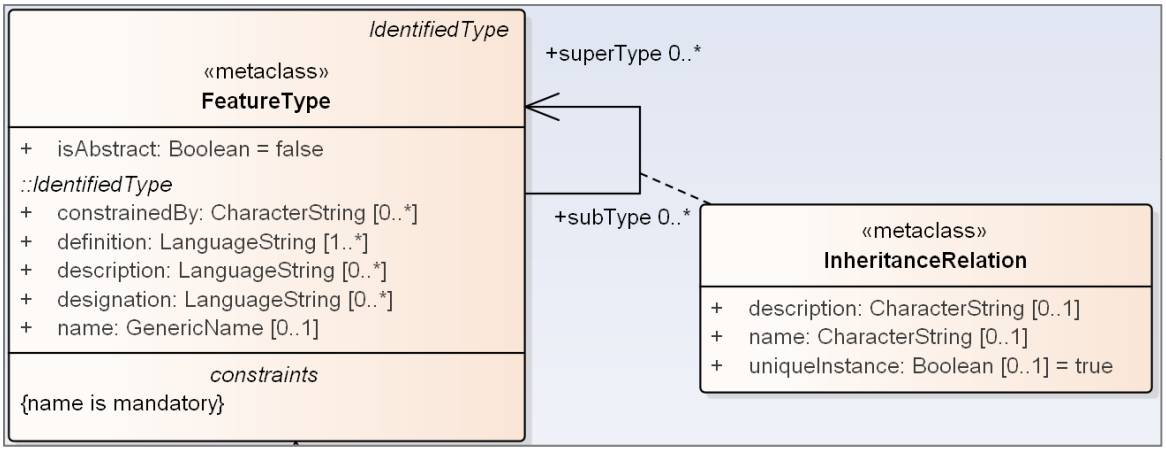

Figure 1 – Entity Generalization in the ISO 19109 General Feature Model

The NTAX information model does not contain representations for all of the GFM modeling concepts. In particular, the NTAX information model selectively represents only the type-generalization concept of the Inheritance Relation in the GFM, shown in Figure 1. The inverse type-specialization concept is implicit in the model and may be logically inferred.

5.2.3 NTAX Information Model Concepts and Definitions

The NTAX information model comprises four main information modeling constructs:

- an ontology class (for representing and documenting the NSG Taxonomy as a whole),
- an entity class (for representing the individual entity types included in the NTAX content),
- an abstract class of properties for documenting the taxonomy and its classes, and
- a class-axiom construct representing the disjointness constraint on collections of sibling subclasses that is specified in the GFM by the Boolean property ‘uniqueInstance’.

This information model is presented diagrammatically in Figure 2.[6]

[6] See Annex D for explanation of the notation.

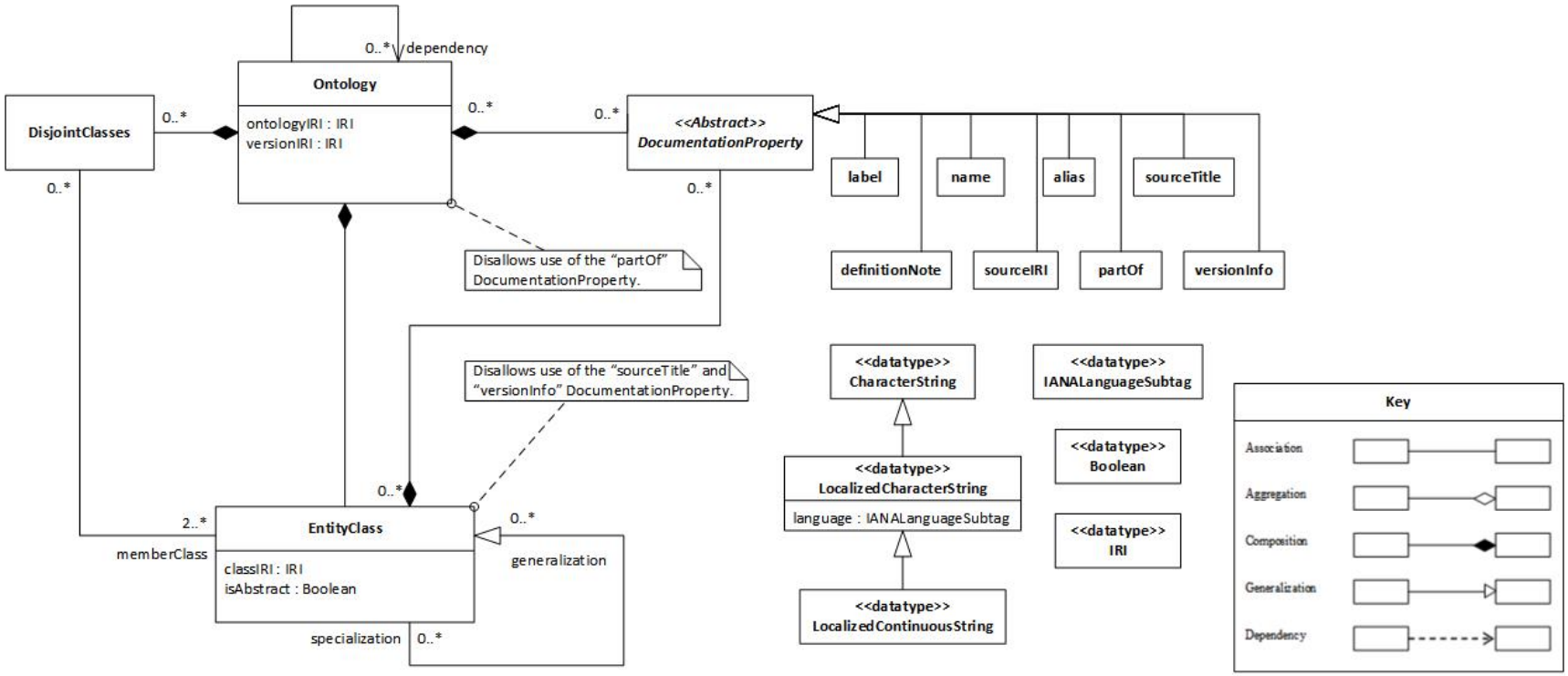

Figure 2 – Overview of the NTAX Information Model

Each element of the NTAX information model is defined in Table 4, below. The table format used to document the information modeling concepts of the NTAX information model is as follows:

- The **Reference** column consists of a sequentially-assigned, non-normative identifier of the element (class or property) that is provided for cross-referencing purposes. It may vary from version-to-version of this document.
- The **NTAX Modeling Concept** column specifies the class name, class attribute name, or class role name of the information modeling concept. For clarity, role names are prefixed by the italicized phrase "*Role name*".
    - A specified class in the model has a capitalized name and always appears in the table on a grey-highlighted row above its properties.
    - The properties (attributes and/or association roles) of a model class are specified in subsequent rows of the table below the class row.
- The **Definition** column specifies the definition of the model class or property.
- The **Source of Definition** column records the source of the definition of the information modeling concept.
- The **Obligation** column specifies if the property is **Mandatory**, Conditional, or *Optional*.
    - Properties whose obligation is "**Mandatory**" shall be populated in accordance with the property definition and any associated guidance.
    - Properties whose obligation is "Conditional" are mandatory when the stated condition is satisfied, in which case they shall be populated in accordance with the property definition and any associated guidance.
    - Properties whose obligation is "*Optional*" are optional, but their population is good business practice when the applicable information is available.
- The **Multiplicity** column indicates the number of instances of the value type of the property that are permitted by this information model. In the case that more than a single domain value of the property is allowed, an indication may also be included in this column if the ordering of the domain values is significant.
- The **Value Type** column indicates the modeling concept or datatype that is used to define the value(s) of the property.

Table 4 – Definitions of Modeling Concepts in the NTAX Information Model

Ref #	NTAX Modeling Concept	Definition	Source of Definition	Obligation	Multiplicity	Value Type
1	Ontology	A formal representation of phenomena of a universe of discourse (4.1.36) with an underlying vocabulary including definitions and axioms that make the intended meaning explicit and describe phenomena and their interrelationships.	ISO 19101-1:2014, 4.1.26			
2	ontologyIRI	A uniform resource identifier (URI) that uniquely identifies an ontology, consisting of a URI base owned by the organization that maintains the ontology, concatenated (following a separator "/") with an abbreviation of the name of the ontology.	ISO 19150-2:2015, 6.2.2	**Mandatory**	Exactly one	IRI
3	versionIRI	A uniform resource identifier (URI) that uniquely identifies a specific version of an ontology, consisting of a URI base owned by the organization that maintains the ontology, concatenated (following a separator "/") with an abbreviation of the name of the ontology and (following a separator "/") with the version indicator (*e.g.*, year or version number).	ISO 19150-2:2015, 6.3.3	**Mandatory**	Exactly one	IRI
4	versionInfo	A character string indicating a unique state in the life of a managed resource (for example: by date or number).	Based on ISO 19135:2005	**Mandatory**	Exactly one	CharacterString
5	*Role name:* dependency	An ontology from which this ontology re-uses one or more concepts.	ISO 19150-2:2015, 6.3.3	*Optional*	Zero or more	Ontology
6	EntityClass	A concept for a set of individual objects that share the same nature and normative properties.	Based on UML 2.4			
7	classIRI	A uniform resource identifier (URI) that uniquely identifies a class, consisting of the ontology IRI concatenated (following a separator) with the label of the class.	ISO 19150-2:2015, 6.2.4	**Mandatory**	Exactly one	IRI
8	isAbstract	A Boolean value indicating that this Entity Type is abstract (and thus not intended to be realized in an implementation).	ISO 19109:2015, 7.4.5 isAbstract	Conditional	If applicable, then exactly one.	Boolean

Ref #	NTAX Modeling Concept	Definition	Source of Definition	Obligation	Multiplicity	Value Type
9	*Role name:* generalization	The relationship between an entity class and its superclass(es), such that all individuals belonging to the subclass also belong to the superclass and satisfy its definition. A subclass inherits the properties of its superclass(es). The superclass is the generalized class, while the subclasses are typically specified with additional properties.	ISO 19109:2015, 7.4.5 superType; 7.4.12 generalization	Conditional	If applicable, then one or more.	EntityClass
10	DocumentationProperty {Abstract}	An abstract modeling entity that is a superclass for properties used to provide a textual note or other information intended to characterize a concept for human understanding.				
11	label	A compressed human-readable designator for a resource that may be used as the terminal segment of its resource IRI. The language may be indicated.	Based on ISO 19135:2005, 7.2 Item identifiers; ISO 19150-2:2015, 6.3.3 and 6.4.3	**Mandatory**	Exactly one	LocalizedContinuousString
12	name	The preferred human-readable designator that is used to denote the concept in a specified language.	ISO 19135:2005, 7.2 Item names, 8.6.2 name	**Mandatory**	Exactly one	LocalizedCharacterString
13	alias	A functionally equivalent synonym for a concept in an alternative context or language.	ISO 19110:2016, Table B.2, No. 2.5 aliases	*Optional*	Zero or more	LocalizedCharacterString
14	definitionNote	A precise statement of the nature and normative properties of a concept, followed by an optional statement about relevant non-essential qualities, variations, scope, and/or examples. The language of the definition may be indicated.	ISO 19135:2005, 7.3 Definitions; ISO 19150-2:2015, 6.4.3	**Mandatory**	Exactly one	LocalizedCharacterString
15	partOf	A related resource in which the described resource is physically or logically included.	DCMI Metadata Terms (2012)	Conditional	If applicable, then exactly one.	IRI
16	sourceIRI	The Internationalized Resource Identifier (IRI) of the recommended reference to be used for information about the concept.	ISO 19115:2003, 2.4.2.3 (line 96), sourceCitation	**Mandatory**	Exactly one	IRI
17	sourceTitle	The title of the ontology reference document or standard on which the ontology is based.	ISO 19150-2:2015, 6.3.3	**Mandatory**	Exactly one	LocalizedCharacterString
18	DisjointClasses	A collection of entity classes, all of which are pairwise disjoint, indicating that no individual can belong at the same time to more than one of the member classes ("types"). For example, a collection of sibling subclasses having the same generalization, where instances of the supertype shall not be an instance of more than one of the subtypes.	ISO 19109:2015, 7.4.12 uniqueInstance			
19	*Role name:* memberClass	An entity class belonging to this collection of disjoint classes.		Conditional	If applicable, then two or more.	EntityClass

5.2.4 Datatypes in the NSG Taxonomy Information Model

5.2.4.1 Introduction

The NTAX information model includes the following datatypes which are specified as value types for properties in the information model.

5.2.4.2 CharacterString

The CharacterString datatype represents a finite-length sequence of zero or more characters from the Universal Character Set (Unicode), as specified by ISO/IEC 10646. The character string may be accompanied by a formal identifier (*i.e.*, a token or "tag") used to identify the natural language of the expression represented by the string. A character string may be further specified, including with respect to length (exact, minimum, or maximum) or pattern (for example, a pattern that excludes space characters).

5.2.4.3 LocalizedCharacterString

The LocalizedCharacterString datatype represents a character string for which the natural language to use in interpreting the content is specified by a language code ("tag"). A character string is a finite-length sequence of zero or more characters from the Universal Character Set (Unicode). A language tag is a lowercase abbreviation for the natural language of the expression represented by a character string.

5.2.4.4 LocalizedContinuousString

The LocalizedContinuousString datatype represents a character string having no whitespace characters (unless they are encoded by '%20') and for which the natural language to use in interpreting the content is specified by a language code ("tag").

5.2.4.5 IANALanguageSubtag

The IANALanguageSubtag codelist represents a set of formal identifiers for natural languages, as defined by BCP 47 (currently represented by RFC 4646 and RFC 4647) or its successor(s). IANA language subtags 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.[7]

5.2.4.6 Boolean

The Boolean datatype represents the values of two-valued logic. A Boolean value is either TRUE or FALSE.

5.2.4.7 IRI

The IRI datatype represents International Resource Identifiers (IRIs). An IRI is a finite-length sequence of characters from the Universal Character Set (Unicode/ISO 10646) that is intended to identify an abstract or physical resource as described in IETF RFC 3987.

NOTE	The NTAX Standard document does not include the specific taxonomy content (*i.e.*, individual entity classes) included in any particular content baseline of the NSG Taxonomy (NTAX). NTAX content is published as technical artifacts (encodings) that may be accessed online in the NSG-unique Standards Register of the NSG Standards Registry. NTAX content is also accessible through the REST API component of the NSG Standards Registry. See Section 6.2.

[7] The complete IANA Language Subtag registry content is available at the following URL: http://www.iana.org/assignments/language-subtag-registry/language-subtag-registry.

5.3 NSG Taxonomy Representation using Semantic Web Languages

5.3.1 The Semantic Web

The information modeling elements of the NTAX information model can be represented using the Web Ontology Language, Second Edition (OWL 2), as defined by W3C Recommendations for the Semantic Web. The use of OWL 2 enables the encoding of NTAX content in machine-readable formats that can be used with Web-based applications and shared on the Web to provide machine-processable semantics for data discovered from multiple sources (for example, in Linked Data).[8]

The Semantic Web is a virtual set of distributed data accessible on the internet that is represented using standards-based, machine-processable descriptions that allow the data to be application-independent and available for re-use in accordance with a framework of common standards.[9] Data in the Semantic Web can be discovered, queried, aggregated, and analyzed as part of the larger information ecosystem by leveraging the semantics (*i.e.*, meanings) of the data. The phrase "the Web of Data" is used synonymously with the Semantic Web in this sense.[10]

The term "Semantic Web" also encompasses the technologies, including the standards and operational infrastructure, that support the creation of the Web of Data. Semantic Web standards define a framework (including representation languages and exchange formats) for describing data in a reusable, machine-processable way, as well as guidelines for creating the operational environment on the Web.[11] Finally, the Semantic Web relies on an implemented technology infrastructure that enables the real-time publication, linking, and processing of data published in Semantic Web exchange formats.

The World Wide Web Consortium (W3C) has developed a set of recommendations (standards) that can be used together with non-W3C standards (such as Unicode) to support the representation and exchange of information on the Web. All of these recommendations depend upon unique identifiers using standardized character sets to identify resources on the Web. Initial efforts to support the sharing of information on the Web through the creation of Extensible Markup Language (XML)[12] focused on the representation of individual data objects (*i.e.*, instance data) together with metadata about them. Subsequently, three W3C recommendations (RDF, RDFS, and OWL) defined new representation languages that are used to formalize the semantics of data and its real-world domains in a machine-processable form. Those recommendations together enable support for automated logical reasoning about the data (*e.g.*, inferencing to check constraints or to conclude additional facts from known data) as well as querying for data across the Web.

The set of standards used to enable sharing of the semantics of information on the Web is often referred to as the "Semantic Web Stack", because later recommendations built upon and extended the capabilities of earlier standards. Figure 3 shows graphically the reliance on and dependencies among the recommendations and standards that are used together to enable the Semantic Web.

The key standards for capturing the semantics of data are RDF, RDFS, and OWL. The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent complex knowledge about entities, groups of entities, and relationships between entities. OWL is a formal language based on computational logic, which allows knowledge expressed to be processed by computer programs, including reasoners.[13]

[8] A similar approach to data definition and data linking may be used in a closed networked system, rather than on the open internet, when required for mission-specific purposes.
[9] *The Semantic Web*. Michael C. Daconta, Leo J. Obrst, and Kevin T. Smith. Wiley Publishing, Inc. 2003. Page 4.
[10] Linked Data Glossary, W3C Working Group Note 27 June 2013 (http://www.w3.org/TR/ld-glossary/)
[11] "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries." (W3C FAQ, *What is the Semantic Web*: http://www.w3.org/2001/sw/SW-FAQ)
[12] XML was developed in the late 1990s to provide a syntax for creating markup languages to capture metadata.
[13] W3C Web Ontology Language (OWL). Accessed online at: http://www.w3.org/OWL/.

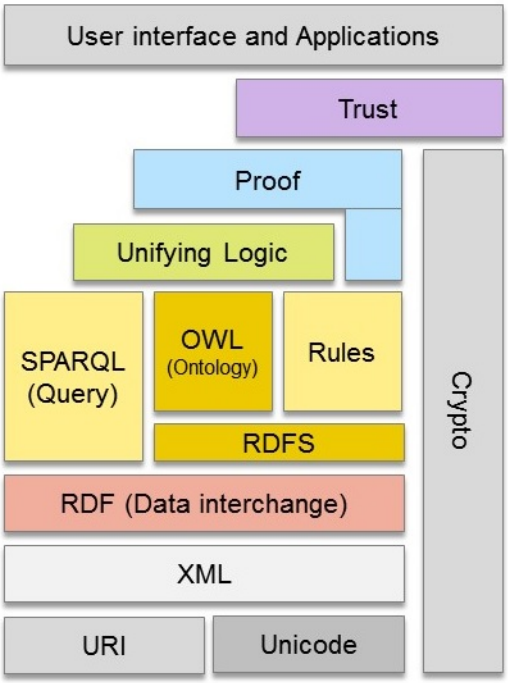

Figure 3 – The Semantic Web Stack[14]

The Semantic Web stack supports a common framework that allows data to be shared and reused across application, enterprise, and community boundaries, through the extraction, representation, storage, retrieval and analysis of machine-processable.[15] They include standards that define exchange formats (primarily RDF) for sharing data on the Web (sometimes called the "Web of Data").

Data that is prepared, published, shared, and consumed in the Semantic Web is called "Linked Data". As defined by the W3C, Linked Data is data that is implemented using:

> A pattern for hyperlinking machine-readable data sets to each other using Semantic Web techniques, especially via the use of RDF and URIs. This enables distributed SPARQL queries of the data sets and a browsing or discovery approach to finding information (as compared to a search strategy). Linked Data is intended for access by both humans and machines. Linked Data uses the RDF family of standards for data interchange (*e.g.*, RDF/XML, RDFa, Turtle) and query (SPARQL).[16]

Guidance on the production and publication of data instances is outside the scope of the NTAX Standard.[17] In January 2017, the W3C Data Activity published "Data on the Web Best Practices", a W3C Recommendation (31 January 2017), available at http://www.w3.org/TR/dwbp/. Topics covered in this publication include: metadata, licenses, provenance, quality, versioning, identifiers, formats, vocabularies, access (APIs), and data preservation.[18]

5.3.2 Selecting OWL Constructs for Representation of NTAX Content

The NTAX content is represented using the W3C Web Ontology Language (OWL 2). The specific variant of OWL 2 used is OWL-DL, which supports tractable reasoning by inference engines.

---

[14] Based on *Semantic Web and Other W3C Technologies to Watch*. Steve Bratt, CEO, W3C. October 2006. Retrieved online at: http://www.w3.org/2006/Talks/1023-sb-W3CTechSemWeb/. There is some variation in depictions of the stack, which has changed over the years with the addition of new recommendations such as the Rules Interchange Format (RIF) and the adoption of IRIs to provide a broader method than URIs for constructing unique identifiers. There are also variations in which the diagram consists of uniform flat layers, although in fact the relationship between layers is more complex than in the stack-of-pancake depictions.

[15] Linked Data Glossary, W3C Working Group Note 27 June 2013. (http://www.w3.org/TR/ld-glossary/)

[16] Linked Data Glossary, W3C Working Group Note 27 June 2013. (http://www.w3.org/TR/ld-glossary/)

[17] The sole way in which the NTAX Standard addresses instance data is in the requirement for classifying instances using the NSG Taxonomy, as specified in Annex A (A.2.4.1)

[18] Guidance from the W3C Data Activity acknowledges the importance of standards to provide semantics for shared data. All semantic resources are presented under the topic of "Vocabularies", including ontologies and taxonomies as well as controlled terminologies.

A taxonomy consists of a hierarchy of entity-class concepts used to categorize phenomena in a universe of discourse, typically based on the presumed properties of real-world entities. A taxonomy may be developed as part of an ontology – specifically, as the hierarchy of classes that is the core structure or "backbone" of the ontology. The NSG Taxonomy is derived from the hierarchy of entity types in the NSG Application Schema (NAS). The relationship between the information content of the NTAX and the NAS is described in Annex C.

The formal representation language used for the NTAX is a subset of the specification for the W3C Web Ontology Language defined in the *OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax*.[19] Elements from OWL 2 are used to represent the structural concepts and properties of the NTAX information model (*i.e.*, classes, subclass relationships, and disjointness). Annotation properties from OWL 2 and other recommendations – including the Simple Knowledge Organization System (SKOS) – are used to incorporate documentation of NAS-derived semantics within the taxonomy components.

The NTAX taxonomy specification subsets the OWL 2 specification in the following ways:

1. OWL Data Property Expressions are excluded.
2. OWL Object Property Expressions are excluded.
3. OWL Class axioms are restricted to SubClassOf and DisjointClasses (*cf.* OWL 2 Structural Specification: Section 9.1).
4. Of the OWL Datatypes, only the RDF PlainLiteral and Boolean are included (OWL 2: Sections 4.3 and 5.7).
5. Only the following three OWL 2 Annotation properties are included in NTAX DocumentationProperty: `rdfs:label`, `rdfs:isDefinedBy`, `owl:versionInfo`.

The NTAX taxonomy specification also reuses several annotation properties from other standards, as listed in Section 5.3.6.

5.3.3 Representing NTAX Information Model Concepts in OWL

The standard ISO 19150-2, *Geographic information – Ontology – Part 2: Rules for developing ontologies in the Web Ontology Language (OWL)*, defines a rule-based transformation from ISO 19100-series compliant UML application schemas to OWL 2 ontologies. The NTAX Standard relies on correlations defined by those transformation rules in order to determine the OWL 2 representation of modeling elements in the NTAX information model.

The rules specified in ISO 19150-2 are the basis for determining the representation for most elements of the NTAX information model in terms of constructs from OWL 2 and other Semantic Web standards, as listed in Table 5, Table 6, and Table 7. Where ISO 19150-2 does not address the OWL representation for an aspect of the ISO 19109 GFM model (especially, the disjointness of sibling subclasses indicated by 'uniqueInstance'), an OWL 2 encoding was developed (see Section 5.4.3.4).

5.3.4 Unique Identifiers in OWL: IRIs

OWL 2 ontologies and ontology elements are identified using Internationalized Resource Identifiers (IRIs). An IRI is a finite-length sequence of zero or more characters used for identifying an abstract or physical resource. A resource can be anything that has identity. IRIs may be used solely for identification of resources, or they may also be used to locate and access resources.

> **NOTE** An Internationalized Resource Identifier is a sequence of characters from the Universal Character Set (Unicode/ISO 10646) that forms an identifier for a resource. IRIs complement an older format, Uniform Resource Identifiers (URIs), which allows the use of only a subset of the ASCII character set to construct identifiers. A standardized mapping from IRIs to URIs is defined in the IRI specification. When a resource identifier is used for resource retrieval, it may be necessary to determine the associated URI, because retrieval mechanisms may be defined only for URIs. Every URI is by definition an IRI.[20]

An encoding of the NSG Taxonomy is a Web resource identified by an IRI that is a URI. In addition, each component in the encoded NTAX content – including its classes, properties, and individuals – is a resource and, as such, is identified by an IRI that is a URI.

[19] Accessed online at: http://www.w3.org/TR/owl2-syntax/.
[20] See: http://tools.ietf.org/html/rfc3987#section-3.1.

The use of URIs for the NSG Taxonomy and its components is consistent with requirements for the identification of resources on the internet. The intent of the World Wide Web is to enable sharing of locatable resources across a global community with both known and unanticipated users. Information-sharing is supported by use of a single global identification system that provides a common basis for unique identification of resources across the Web. Identification of Web resources by IRIs and URIs is a recommended practice of the World Wide Web Consortium (W3C). Benefits of using URIs to locate resources include linking, caching, bookmarking, and indexing by search engines. Key to the use of URIs is that each URI should identify a single distinct resource.[21]

5.3.5 Structural Elements of the NSG Taxonomy

The structural elements of the NSG Taxonomy are its entity classes, generalization relationships, and subclass-disjointness axioms.

The NSG Taxonomy as a whole is represented as an instance of `owl:Ontology`.

Entity classes in the NTAX content are represented by instances of `owl:Class`.

Generalizations in the NTAX content are represented by the property `rdfs:subClassOf`. The NTAX information model represents only the type-generalization direction of the Inheritance Relation in the ISO 19109 GFM, shown in Figure 1. The inverse type-specialization concept is logically implicit in the interpretation of the subclass property and may be logically inferred.

Information about disjoint sibling subclasses is indicated in the ISO 19109 GFM by the Boolean property 'uniqueInstance'. This is represented in the encoded NTAX content by OWL class axioms using `owl:AllDisjointClasses` to enumerate the classes that are disjoint. Disjoint classes have no individual members in common. Section 5.4 contains more information about the implementation of encodings for disjoint-classes axioms.

5.3.6 Documentation Properties in the NSG Taxonomy

The encoded NTAX content includes information intended for human consumption as well as for machine reasoning. This documentation is represented using annotation properties for OWL 2 (listed in Section 5.3.2 above) and several other information standards. The annotation properties used for NTAX documentation are presented below, grouped by standard. Their specific use with NTAX components is explained in the presentation of the NTAX encoding.

- **OWL**
  - `owl:versionInfo`
- **RDF Schema**
  - `rdfs:label`
  - `rdfs:isDefinedBy`
- **SKOS**
  - `skos:altLabel`
  - `skos:definition`
  - `skos:prefLabel`
- **Dublin Core Terminology**
  - `dct:isPartOf`[22]
  - `dct:source`

The documentation properties are applied to the NSG Taxonomy as a whole and also to the component classes in the NTAX content, as specified in the NTAX information model presented in Section 5.2.3.

[21] Resources are broadly inclusive of Web pages, images, concepts, and even real-world objects. *Architecture of the World Wide Web, Volume One*. W3C Recommendation 15 December 2004. Ian Jacobs and Norman Walsh, Eds. Available online at: http://www.w3.org/TR/webarch/#identification.

[22] This property is used only in the publication of individual ontology resources (see Section **Error! Reference source not found.**).

5.4 NSG Taxonomy Encodings

5.4.1 Introduction

The NTAX Standard specifies two technology-specific encodings of the NTAX content that each accurately represents the information model in the NTAX Standard:

- RDF/XML encoding – RDF/XML is the primary concrete exchange syntax for OWL 2, as specified in the W3C Recommendations. All OWL 2 tools are required to support the OWL 2 RDF/XML syntax (see Section 2.1 of the OWL 2 Conformance document[23]).
- N-Triples – N-Triples is a line-based, plain-text format for encoding an RDF graph that may be used in information exchange without necessitating the complicated parsing required for RDF/XML. In N-Triples, each line consists of a sequence of three RDF terms representing, respectively, the Subject, Predicate, and Object of an RDF triple.[24]

Some aspects of encoding the NTAX content are the same for both technologies.

The following sections present the general aspects of the NTAX encodings first, followed by specific differences for the RDF/XML and N-Triples encodings. Both encodings use IRIs as described in Section 5.4.2 to uniquely identify model elements. Section 5.4.3 describes the general approach for encoding all information modeling concepts of the NTAX information model in OWL 2. Section 5.4.5 lists the differences between RDF/XML and N-Triples encodings of the NTAX content.

The RDF/XML encoding employs the approach defined in ISO 19150-2:2015, as implemented and extended in the OGC Testbed-12 ShapeChange Engineering Report (2016). The N-Triples encoding is derived from the RDF/XML encoding.

5.4.2 Namespace and Identifiers

5.4.2.1 Introduction

The NSG Taxonomy is a resource that includes a collection of entity classes, which are also resources. In the World Wide Web, resources must be uniquely identified by IRIs. Related resources may be grouped together into a "namespace" using a specified IRI structure for all resources in the namespace.

5.4.2.2 Namespace

A namespace identifies a collection of resources by referencing them using identifiers (IRIs) that share a common initial prefix or "stem" (also referred to as a URI base). An RDF namespace is represented by a URI base used in all identifiers for a set of related resources. The namespace URI base is concatenated with a separator followed by a local name to create the complete IRI identifier for an individual RDF resource.[25] The encoding of NTAX content uses the "/" separator for this purpose.

Each IRI belongs to a single namespace. Resources from different namespaces may be combined in the construction of a new resource. The NTAX content encodings re-use elements from the Web Ontology Language, the Resource Description Framework, and other Web resources. Every modeling element from RDF, RDFS, OWL, and SKOS that is re-used in encodings of NTAX content has a unique IRI that identifies that element in relation to its namespace. For example, the OWL concept Class (which has the IRI http://www.w3.org/2002/07/owl#Class) is in the OWL namespace (http://www.w3.org/2002/07/owl#).

Components of the NSG Taxonomy have a URI base that identifies them as belonging to the NTAX namespace, as described in the next section.

> **NOTE** The NTAX Standard and NTAX content in RDF/XML use prefix abbreviations for common namespaces as declared in the OWL Structural Specification (Section 2.4); *e.g.*, "owl" for the namespace identified by http://www.w3.org/2002/07/owl#. NTAX content in N-Triples uses only fully specified namespace prefixes.

[23] OWL 2 Web Ontology Language Conformance (Second Edition). W3C Recommendation. 11 December 2012. Michael Smith, *et al*., eds. Published online at: http://www.w3.org/TR/2012/REC-owl2-conformance-20121211/.

[24] RDF 1.1. N-Triples. W3C Recommendation. 25 February 2014. David Beckett. Published online at: http://www.w3.org/TR/n-triples/.

[25] In the concatenation of a URI base with a local name, a separator (which may be either the hash ("#") or the forward slash ("/")) character is required between the two parts. The type of separator used depends upon the supported retrieval mechanism.

5.4.2.3 NTAX Identifiers

The NSG Taxonomy has a unique identifier for the taxonomy (as a whole) as a complete resource. Each entity class belonging to the NTAX namespace also has a unique identifier, which embeds the identifier of the NSG Taxonomy as the URI base (‘http://api.nsgreg.nga.mil/taxonomy/ntax’). An NTAX entity-class identifier is the combination of the URI base and (following the appropriate separator) a unique terminal label for the concept (*e.g.*, ‘Building’, ‘MountainPass’).

NTAX IRIs are in the form of a Uniform Resource Locator (URL). A URL specifies the location of, and access to, a resource on the Internet. A URL specifies the protocol of the resource (*e.g.*, 'http' or 'ftp'), the domain name for the resource (*e.g.*, 'nsgreg.nga.mil'), and the relative location of the resource within that domain. If the site host is active, then accessing the specified resource results in retrieval of a representation (*i.e.*, the content) of the resource; however, site persistence is not guaranteed.

5.4.2.4 Versioned and Non-versioned IRIs for NTAX

The NTAX Standard includes both versioned and non-versioned IRIs for the NSG Taxonomy. The versioned IRI shall be used for authoritative identification of the NSG Taxonomy (as a whole) and NTAX content in information exchange and data sharing. The versioned IRI shall also be used for official specification of the NSG Taxonomy version to be used in systems development or acquisition.

- Example of NTAX versioned IRI: http://api.nsgreg.nga.mil/taxonomy/ntax/9.0

For convenience, the REST API component of the NSG Standards Registry supports the retrieval of the current version of the NSG Taxonomy when the non-versioned IRI is used.

- Example of NTAX non-versioned IRI: http://api.nsgreg.nga.mil/taxonomy/ntax

5.4.2.5 Form of IRIs for NTAX Content

IRI component-designations for NTAX content are constructed in accordance with the following pattern:

protocol ":/\/" *domain* "/" *resource-type* "/" *resource* "/" *version* "/" *concept*

Each component in the pattern is case-sensitive and determined as follows:

- *protocol* – always 'http'
- *domain* – always 'api.nsgreg.nga.mil'
- *resource-type* – always 'taxonomy'
- *resource* – always 'ntax'
- *version* – designates the version of the resource (*e.g.*, '9.0', 'base17Feb')
- *concept* – designates an individual concept (*e.g.*, 'Aerial', 'Building', 'MountainPass')

The individual components are concatenated into a single string as specified by the pattern (above), to form the IRI that designates the associated taxonomy component. For example:

http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Aerial	(The OWL class representing the entity class 'Aerial' in the NSG Taxonomy content, Version 9.0)
http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Building	(The OWL class representing the entity class 'Building' in the NSG Taxonomy content, Version 9.0)
http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/MountainPass	(The OWL class representing the entity class 'MountainPass' in the NSG Taxonomy content, Version 9.0)

The IRI for a specific entity class in the NTAX may be used to access the individual resource (in RDF/XML or N-Triples encoding) for that entity class from the REpresentational State Transfer (REST) API component of the NSG Standards Registry.

5.4.3 General NTAX Encoding

5.4.3.1 Introduction

Section 5.2.3 of the NTAX Standard identifies the four basic types of information modeling concept in the NTAX information model: Ontology, EntityClass, DocumentationProperty, and DisjointClasses. The class DocumentationProperty is an abstract class, which is not represented in the encoding. The documentation properties are assigned to their appropriate modeling concepts in Table 5, Table 6, and Table 7.

The Ontology and each EntityClass concept have several types of information specified, including required and optional attribution. The IRI-valued attributes provide identification of the modeling concepts. The remaining attribution provides the means of further specifying an information modeling element by documentation properties and structural relationships (*e.g.*, entity-class generalization).

Encoding elements for the NTAX information model components are defined in Table 4. The table format used to document these encoding elements is as follows:

- The **NTAX Modeling Concept** column specifies the class name, class attribute name, or class role name of the information modeling concept.
- The **NTAX Encoding Element** column specifies the OWL 2 or other Semantic-Web standard construct that shall be used to represent the corresponding NTAX Modeling Concept. NTAX encoding elements are shown in the form used by the (normative) RDF/XML encoding of NTAX content. This includes the use of namespace abbreviations rather than fully specified IRIs (which are used by the N-Triples encoding).
- The **Cardinality of Element** column indicates the number of occurrences of the element that are permitted by the information model.
- The **Value Type** column indicates the modeling concept or datatype that is used to define the value(s) of the element.
- The **Notes** column contains comments, specifications of the actual value, or examples of values for the element.

5.4.3.2 Encoding of the Ontology Representing the Taxonomy

The Ontology modeling concept is used to represent the NSG Taxonomy itself as a self-documenting resource. Each encoding of the NTAX is an Ontology characterized by properties as specified in Section 5.2.3. The NTAX information model elements for the NSG Taxonomy are encoded in OWL 2 and supporting W3C languages as specified in Table 5.

5.4.3.3 Encoding of the Entity Class concept

The EntityClass modeling concept is used to represent sets of individual objects that share the same nature and normative properties. An EntityClass is characterized by the following properties as specified in Section 5.2.3. The NTAX Information Model elements for an EntityClass are encoded in OWL 2 and supporting W3C languages as specified in Table 6.

Table 5 – NTAX Encoding Elements for the Taxonomy

NTAX Modeling Concept	NTAX Encoding Element	Cardinality of Element	Value Type	Notes
Ontology	`owl:Ontology`			Used to represent the NSG Taxonomy
ontologyIRI	`rdf:about`	Exactly one	IRI	For example: http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb
versionIRI	`owl:versionIRI`	Exactly one	IRI	For example: http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb
versionInfo	`owl:versionInfo`	Exactly one	CharacterString	For example: “base17Feb”
label	`rdfs:label`	Exactly one	LocalizedContinuousString	The value is “NsgTaxonomy”.
name	`skos:prefLabel`	Exactly one	LocalizedCharacterString	The value is “NSG Taxonomy”.
alias	`skos:altLabel`	One or more	LocalizedCharacterString	For example: “NTAX”
definitionNote	`skos:definition`	Exactly one	LocalizedCharacterString	The definition note for the NTAX is: “The NSG Taxonomy (NTAX) Standard defines a logic-based taxonomy specification in the W3C Web Ontology Language (OWL 2) for a generalization (*i.e.*, subclass) hierarchy of entity classes representing concepts used for the typing of Geospatial Intelligence (GEOINT) information shared in the U.S. National System for Geospatial Intelligence (NSG). Description: The content of the NSG Taxonomy contains a computer-interpretable representation of the subclass hierarchy of entity classes derived from the NSG-wide logical model for geospatial data (*i.e.*, the NSG Application Schema (NAS)), which is implemented in two types of OWL 2 encodings: RDF/XML and N-Triples. The NSG Taxonomy supports the categorization of NSG instance data, and also supports indexing, navigation, search, and retrieval when used in conjunction with suitable software applications supporting W3C standards.”
sourceIRI	`rdfs:isDefinedBy`	Exactly one	IRI	For the NTAX Standard Edition 1.0, the URL is: http://nsgreg.nga.mil/doc/view?i=2616
sourceTitle	`dct:source`	Exactly one	LocalizedCharacterString	The value is “NSG Taxonomy (NTAX) Standard”.
Role name: dependency	`owl:imports`	Zero or more	Ontology	For example: http://www.w3.org/2004/02/skos/core# (SKOS)

Table 6 – NTAX Encoding Elements for an Entity Class

NTAX Modeling Concept	NTAX Encoding Element	Cardinality of Element	Value Type	Notes
EntityClass	`owl:Class`			Used to represent NSG Taxonomy entity classes
classIRI	`rdf:about`	Exactly one	IRI	For example: http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome
isAbstract	`iso19150-2:isAbstract`	If applicable, then exactly one.	Boolean	FALSE (By policy: FALSE, unless asserted as TRUE.)
label	`rdfs:label`	Exactly one	LocalizedContinuousString	For example: 'LandAerodrome'
name	`skos:prefLabel`	Exactly one	LocalizedCharacterString	For example: 'Land Aerodrome'
alias	`skos:altLabel`	One or more	LocalizedCharacterString	For example: 'Airport'
definitionNote	`skos:definition`	Exactly one	LocalizedCharacterString	For example: (Land Aerodrome) "Definition: An aerodrome on land intended to be used either wholly or in part for the arrival, departure and surface movement of aircraft. Definition: [None Specified]"
sourceIRI	`rdfs:isDefinedBy`	Exactly one	IRI	For example: http://nsgreg.nga.mil/as/view?i=100436[26]
partOf	`dct:isPartOf`	If applicable, then exactly one.	IRI	For example: http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb (the IRI for the NSG Taxonomy as a whole)
Role name: generalization	`rdfs:subClassOf`	If applicable, then one or more.	EntityClass	For example: (Maritime Vessel) http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Vehicle

[26] The EntityClass IRI incorporates the numerical Item Identifier value for the corresponding NAS Entity Type.

5.4.3.4 Encoding of the Disjoint Classes Concept

The DisjointClasses modeling concept is used to represent collections of entity classes that are pairwise disjoint; that is, no individual can belong at the same time to more than one of the member classes in a specific collection. This concept represents the constraint that sibling subclasses in the taxonomy are disjoint. Each DisjointClasses collection is characterized by the following property, defined in Section 5.2.3, which identifies two or more entity classes as members of the disjoint-class collection.

The encoding of NTAX content uses `owl:AllDisjointClasses` to represent all class axioms declaring class disjointness.[27]

Table 7 – NTAX Encoding Elements for Disjoint Classes

NTAX Modeling Concept	NTAX Encoding Element	Cardinality of Element	Value Type
DisjointClasses	`owl:AllDisjointClasses`		
Role name: memberClass	`owl:members`	If applicable, then two or more.	EntityClass

An axiom stating the pairwise disjointness of a set of entity classes is asserted using the property 'memberClass' to identify each entity class in the collection of entity classes. In the NTAX content, these collections contain a set of sibling subclasses, which must not share members.

Both encodings of NTAX content represent DisjointClasses using the encoding elements above. There are some differences in the implementation-specific encodings, however, which are discussed in Section 5.4.5.

5.4.4 General Encoding of Datatypes

The datatypes included in the NTAX information model are represented as follows:

- The datatype IRI is represented using the XML datatype `xsd:anyURI`, as specified in the OWL 2 Structural Specification and Functional-Style Syntax.
- The datatype Boolean is represented using the XML datatype for OWL 2, `xsd:boolean`, as specified in the OWL 2 Structural Specification and Functional-Style Syntax.
- The datatype CharacterString is represented using the `rdf:PlainLiteral` datatype provided in OWL 2 for the representation of strings optionally with an identified natural language, as specified in the OWL 2 Structural Specification and Functional-Style Syntax: Section 4.3. The `rdf:PlainLiteral` value may be either a character string (`xsd:String`), or an ordered pair consisting of a character string (`xsd:String`) and a lower-case language tag (*e.g.*, "en" for English).[28]
- The datatype LocalizedCharacterString is represented using the `rdf:PlainLiteral` datatype provided in OWL 2 for the representation of strings with the constraint that a language tag identifying the natural language of the content of the string is required.
- The datatype LocalizedContinuousString is represented using the `rdf:PlainLiteral` datatype provided in OWL 2 for the representation of strings, with the constraint that the string must not contain spaces (unless encoded by '%20') and with the constraint that a language tag identifying the natural language of the content of the string is required.
- The values of the datatype IANALanguageSubtag are represented using the two-character, lowercase language abbreviations specified in BCP 47.

[27] The NSG Taxonomy uses `owl:AllDisjointClasses` to represent all class axioms declaring class disjointness between two or more classes. The OWL 2 Functional Syntax maps OWL DisjointClasses to RDF graphs using `owl:disjointWith` for two classes, and `owl:AllDisjointClasses` for three or more classes. The OWL Primer illustrates use of `owl:AllDisjointClasses` for the 2-class case. RDF graphs containing `owl:AllDisjointClasses` constructs with two or more classes are mapped to DisjointClasses in the OWL 2 Functional syntax. [OWL 2 Mapping to RDF, Section 3.2.5]

[28] W3C. *rdf:PlainLiteral: A Datatype for RDF Plain Literals*. 11 December 2012. Available online at: http://www.w3.org/TR/2012/REC-rdf-plain-literal-20121211/#Definition_of_the_rdf:PlainLiteral_Datatype.

5.4.5 Technology-specific NTAX Encodings

5.4.5.1 Introduction

The NTAX Standard defines two technology-specific encodings of the NSG Taxonomy conformant with the information model of the NTAX Standard:

- RDF/XML encoding – RDF/XML is the primary concrete exchange syntax for OWL 2. All OWL 2 tools are required to support the OWL 2 RDF/XML syntax (see Section 2.1 of the OWL 2 Conformance document[29]).
- N-Triples – A line-based, plain-text format for encoding an RDF graph.[30]

Each encoding of NTAX content provides a machine-interpretable OWL 2 representation of the entity-class generalization hierarchy for use in storing and exchange of geospatial information in Semantic Web applications. The NTAX content in RDF/XML may be used by Semantic Web tools to enhance search or retrieval of instance data. Data instance files in N-Triples encoding may be linked to related content in the N-Triples NTAX encoding to provide semantics to that data when exchanging it among information systems or making it available in Linked Data stores.

The publication of the NTAX content encodings and how to obtain them is described in Section 6.2.

5.4.5.2 RDF/XML Encoding

The NTAX Standard specifies a technology-specific encoding for the NTAX information model using the mandatory RDF/XML encoding of OWL 2.

- In the RDF/XML encoding, character strings in the 'definitionNote' (`skos:definition`) are encoded using the XML CDATA wrapper.[31]
- In the RDF/XML encoding, where language tags are required or permitted, they shall be provided as the value of an RDF/XML annotation element (`xml:lang`) for the string-valued property, in order to indicate that the content is in English (language code "en").
- In OWL encoded in RDF/XML, assertions about disjoint sibling subclasses are expressed using the class expression `owl:AllDisjointClasses` with a list of the disjoint classes. The encoding specified in Section 5.4.3.4 is used for all collections of disjoint classes, whether there are two or more disjoint classes.

```
<owl:Class rdf:about="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Aerodrome">
    <rdfs:subClassOf
rdf:resource="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/FeatureEntity"/>
    <iso19150-2:isAbstract rdf:datatype=http://www.w3.org/2001/XMLSchema#boolean>true</iso19150-
2:isAbstract>
    <skos:prefLabel xml:lang="en">Aerodrome</skos:prefLabel>
    <rdfs:isDefinedBy rdf:resource="http://nsgreg.nga.mil/as/view?i=103150"/>
    <skos:definition xml:lang="en">![CDATA[Definition: A defined area on land or water (including
any buildings, installations and equipment) intended to be used either wholly or in part for the
arrival, departure and surface movement of aircraft.  Description: [None
Specified]]]</skos:definition>
    <rdfs:label xml:lang="en">Aerodrome</rdfs:label>
  </owl:Class>

  <owl:AllDisjointClasses>
    <owl:members rdf:parseType="Collection">
      <owl:Class rdf:about="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Heliport"/>
      <owl:Class rdf:about="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome"/>
      <owl:Class rdf:about="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/WaterAerodrome"/>
    </owl:members>
  </owl:AllDisjointClasses>
```

Figure 4 – OWL2 RDF/XML Encoding: Entity Class `Aerodrome` and its Disjoint Subclasses

[29] OWL 2 Web Ontology Language Conformance (Second Edition). W3C Recommendation. 11 December 2012. Michael Smith, *et al*., eds. Published online at: http://www.w3.org/TR/2012/REC-owl2-conformance-20121211/.
[30] RDF 1.1. N-Triples. W3C Recommendation. 25 February 2014. David Beckett. Published online at: http://www.w3.org/TR/n-triples/.
[31] In XML the CDATA wrapper is used to indicate to parsers that the enclosed content should not be further interpreted; this allows applications to use characters in data exchange that would otherwise be misinterpreted as element or entity markup.

```
<owl:Class rdf:about="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Heliport">
    <rdfs:subClassOf rdf:resource="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Aerodrome"/>
    <skos:prefLabel xml:lang="en">Heliport</skos:prefLabel>
    <rdfs:isDefinedBy rdf:resource="http://nsgreg.nga.mil/as/view?i=100442"/>
    <skos:definition xml:lang="en">![CDATA[Definition: An aerodrome intended to be used for the arrival, landing, takeoff or departure
of vertical takeoff and landing aircraft/helicopters.  Description: [None Specified]]]</skos:definition>
    <rdfs:label xml:lang="en">Heliport</rdfs:label>
  </owl:Class>

<owl:Class rdf:about="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Aerodrome"/>
    </rdfs:subClassOf>
    <skos:altLabel xml:lang="en">Airfield</skos:altLabel>
    <skos:altLabel xml:lang="en">Airport</skos:altLabel>
    <skos:prefLabel xml:lang="en">Land Aerodrome</skos:prefLabel>
    <rdfs:isDefinedBy rdf:resource="http://nsgreg.nga.mil/as/view?i=100436"/>
    <skos:definition xml:lang="en">![CDATA[Definition: An aerodrome on land intended to be used either wholly or in part for the
arrival, departure and surface movement of aircraft.  Description: [None Specified]]]</skos:definition>
    <rdfs:label xml:lang="en">LandAerodrome</rdfs:label>
  </owl:Class>

<owl:Class rdf:about="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/WaterAerodrome">
    <rdfs:subClassOf rdf:resource="http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Aerodrome"/>
    <skos:altLabel xml:lang="en">Sea Plane Base</skos:altLabel>
    <skos:prefLabel xml:lang="en">Water Aerodrome</skos:prefLabel>
    <rdfs:isDefinedBy rdf:resource="http://nsgreg.nga.mil/as/view?i=100452"/>
    <skos:definition xml:lang="en">![CDATA[Definition: An aerodrome intended to be used either wholly or in part for the arrival,
departure and surface movement of aircraft on water.  Description: [None Specified]]]</skos:definition>
    <rdfs:label xml:lang="en">WaterAerodrome</rdfs:label>
  </owl:Class>
```

Figure 5 – OWL 2 RDF/XML Encoding: Subclasses of `Aerodrome`

5.4.5.3 N-Triples Encoding

The NTAX Standard specifies a technology-specific encoding for the NTAX information model using N-Triples. The W3C Recommendation RDF 1.1 N-Triples specifies a line-based, plain text format for encoding an RDF graph with each triple presented on a separate line followed by a period. N-Triples files do not contain special parsing instructions.

The encoding of NTAX content in N-Triples closely follows the general encoding for the NTAX information model, with the following technology-specific encodings applied:

- Namespace abbreviations are not used in the N-Triples encoding; instead, fully-specified IRIs are used. For example: http://www.w3.org/2002/07/owl#AllDisjointClasses.
- In the N-Triples encoding, the first component (subject) of the triple corresponds to the value of the `rdf:about` in the RDF/XML encoding.
- In the N-Triples encoding, the XML CDATA wrapper is not used (*e.g.*, with the NTAX modeling element ‘definitionNote’).
- In the NTAX N-Triples encoding, where language tags are required or permitted, they shall be appended to the character string by using the ‘@’ symbol, in order to indicate that their content is in English (language code “en”).
- In the N-Triples encoding, DisjointClasses assertions are expressed using the class expression `owl:AllDisjointClasses` with a list of its member entity classes.
- The encoding of `owl:AllDisjointClasses` in OWL N-Triples results in the use of blank nodes to represent: (1) an instance of `owl:AllDisjointClasses`, and (2) the declarations of each member of the list of the disjoint classes. A Skolemized IRI is substituted as the identifier for the list of disjoint classes.

A blank node is a node in an RDF graph that has no IRI identifier. Blank nodes have labels beginning with an underscore character followed by a colon (“_:”). These are not IRIs and cannot be referenced outside of the local graph.

For some applications, it is valuable to assign a unique identifier to the `owl:AllDisjointClasses` construct. This is accomplished through a process termed “Skolemization”. In order to reference a blank node, the label for that node is replaced with a new, skolemized, globally unique IRI corresponding to the blank node. In the NSG Taxonomy, Skolemized IRIs are character strings beginning with the URI base 'http://api.nsgreg.nga.mil/.well-known/genid/', followed by a Universally Unique Identifier (UUID). Other blank nodes used to represent the members of the disjoint-classes construct are not skolemized. See Figure 7 for an example of N-Triples encoding of disjoint subclasses using blank nodes and Skolemized IRIs.

```
<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2002/07/owl#Class> .

<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> <http://www.w3.org/2000/01/rdf-schema#subClassOf>
<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Aerodrome> .

<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> <http://www.w3.org/2004/02/skos/core#altLabel>
"Airfield"@en .

<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> <http://www.w3.org/2004/02/skos/core#altLabel>
"Airport"@en .

<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> <http://www.w3.org/2004/02/skos/core#prefLabel>
"Land Aerodrome"@en .

<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> <http://www.w3.org/2000/01/rdf-schema#isDefinedBy>
<http://nsgreg.nga.mil/as/view?i=100436> .

<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> <http://www.w3.org/2004/02/skos/core#definition>
"Definition: An aerodrome on land intended to be used either wholly or in part for the arrival, departure and surface
movement of aircraft.  Description: [None Specified]"@en .

<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> <http://www.w3.org/2000/01/rdf-schema#label>
"LandAerodrome"@en .
```

Figure 6 – N-Triples Encoding: Entity Class `LandAerodrome`

```
<http://api.nsgreg.nga.mil/.well-known/genid/2F0369E712864B0F9A76B91C650E619D> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2002/07/owl#AllDisjointClasses> .

<http://api.nsgreg.nga.mil/.well-known/genid/2F0369E712864B0F9A76B91C650E619D> <http://www.w3.org/2002/07/owl#members>
_:GCSR7AB2D77100804C3EA1F2962AEC5CF4F4 .

_:GCSR7AB2D77100804C3EA1F2962AEC5CF4F4 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/Heliport> .

_:GCSR7AB2D77100804C3EA1F2962AEC5CF4F4 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>
_:GCSR97EDAE18891C41F59D005997BD82FD9A .

_:GCSR97EDAE18891C41F59D005997BD82FD9A <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/LandAerodrome> .

_:GCSR97EDAE18891C41F59D005997BD82FD9A <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>
_:GCSR05E1A808C50F4766A539DA81B44ED065 .

_:GCSR05E1A808C50F4766A539DA81B44ED065 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/WaterAerodrome> .

_:GCSR05E1A808C50F4766A539DA81B44ED065 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/22-rdf-syntax-nil> .
```

Figure 7 – N-Triples Encoding: Disjoint Subclasses of `Aerodrome`

6 Governance

6.1 Introduction

The management of this NTAX Standard conforms to the governance process established by NGA and executed by the Geospatial Intelligence (GEOINT) Content Standards Board (GCSB). The scope, roles, and governance process of the GCSB are specified in the *GEOINT Content Standards Board (GCSB) Operations Guide*. Changes to the NTAX Standard and its associated NTAX content shall conform to that process. The GCSB is the community forum responsible for providing governance, community coordination, prioritization of content development, and notifications for the set of NGA-developed GEOINT Data Standards that define a common method for specifying and encoding geospatial intelligence and related geospatial information in the National System for Geospatial Intelligence (NSG).

The NTAX Standard and NTAX content evolve in response to GCSB management. The GCSB is responsible for approving changes, distributing change notifications, and publishing the NTAX Standard and the associated technical artifacts and other online resources containing the encodings of NTAX content for use by the U.S. Department of Defense (DoD), U.S. Intelligence Community (IC), and U.S. civil federal agencies. The NTAX Standard and its associated technical artifacts are published in the NSG-unique Standards Register of the NSG Standards Registry. NTAX content is also accessible through the REpresentational State Transfer (REST) API component of the NSG Standards Registry.

6.2 Official Publication of the NTAX Standard and NTAX Content

6.2.1 Introduction

The NTAX Standard and its associated NTAX content shall be published in accordance with the general process described in the GCSB Operations Guide, Section 2.3.5 (Implementation of Changes).

The specific managed content of the NSG Taxonomy consists of:

a) technical artifacts published in the NSG-unique Standards Register of the NSG Standards Registry, and

b) online resources retrievable through the REST API component of the NSG Standards Registry.

All official publications of NTAX content shall conform to the information model specified in Section 5.2 of this NTAX Standard. All encodings shall conform to the OWL 2 representation and encoding for the NSG Taxonomy, as specified in Sections 5.3 and 5.4.

The NTAX Standard itself is published in the NSG-unique Standards Register of the NSG Standards Registry.

6.2.2 Publication of NTAX Content as a Technical Artifact

The NSG Taxonomy technical artifacts are registered files containing the NTAX content authorized for use as of a specified NTAX standard edition or intermediate update, and formatted in encodings as specified in Section 5.4 of this standard. For each such content baseline of the NSG Taxonomy, two files encoding the NTAX content are published, one in OWL 2 RDF/XML and one in OWL 2 N-Triples format. Each file contains the complete content of the NSG Taxonomy for a specified baseline (*i.e.*, a versioned “ntax” namespace). These encoding files are published in the NSG-unique Standards Register of the NSG Standards Registry. A list of content baselines with versioned IRIs for their encoded content files is provided at http://nsgreg.nga.mil/ntax. The latest content baseline is always available at the following non-versioned URL: http://api.nsgreg.nga.mil/taxonomy/ntax. HTTP content negotiation based on the Accept request-header field may be used to specify the media type (as 'application/rdf+xml' or 'application/n-triples') when resource retrieval is requested (the RDF/XML media type is the default).

These encoding files are designed to be machine-processable semantic resources. Their content may be directly examined in a text editor; however, an ontology application with a graphical user interface may be used to inspect the contents of the NSG Taxonomy in a more human-friendly manner. A sample use of an ontology viewer is presented in Annex B.

6.2.3 Publication of NTAX Content as REST API-accessible Resources

Encodings of the NTAX content are also published as a set of online resources that may be accessed through the REST API component of the NSG Standards Registry. These Web resources represent individual entity classes in the NSG Taxonomy. HTTP content negotiation based on the Accept request-header field may be used to specify the media type (as 'application/rdf+xml' or 'application/n-triples') when resource retrieval is requested.

Each such individual resource file shall include a `dct:isPartOf` assertion that links the individual resource to the appropriately versioned NSG Taxonomy namespace. Each individual resource file representing an entity class that has sibling classes shall contain a DisjointClasses assertion declaring the disjointness of the sibling classes. An example of an individual-resource file for ‘Hulk’ (encoded in N-Triples) is presented in Figure 8.

```
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Hulk> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2002/07/owl#Class> .
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Hulk> <http://www.w3.org/2000/01/rdf-schema#subClassOf>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/MaritimeVessel> .
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Hulk> <http://www.w3.org/2004/02/skos/core#prefLabel> "Hulk"@en .
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Hulk> <http://www.w3.org/2000/01/rdf-schema#isDefinedBy>
<http://nsgreg.nga.mil/as/view?i=100279> .
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Hulk> <http://www.w3.org/2004/02/skos/core#definition> "Definition: A vessel, either
stranded aground or permanently moored, that is no longer seaworthy due to an inoperable propulsion plant or compromised vessel
integrity.  Description: [None Specified]"@en .
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Hulk> <http://www.w3.org/2000/01/rdf-schema#label> "Hulk"@en .
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Hulk> <http://purl.org/dc/terms/isPartOf> <http://api.nsgreg.nga.mil/taxonomy/ntax/9.0> .
<http://api.nsgreg.nga.mil/.well-known/genid/E48DF9092DFD40239BF3AD1E91714A6D> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2002/07/owl#AllDisjointClasses> .
<http://api.nsgreg.nga.mil/.well-known/genid/E48DF9092DFD40239BF3AD1E91714A6D> <http://www.w3.org/2002/07/owl#members>
_:GCSR44E72A59F65B4708A7F9D28FCEB30A32 .
_:GCSR44E72A59F65B4708A7F9D28FCEB30A32 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/AmphibiousTransporter> .
_:GCSR44E72A59F65B4708A7F9D28FCEB30A32 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSRCC3D48C26F7444A6965C0AA40EFEF7CD .
_:GCSRCC3D48C26F7444A6965C0AA40EFEF7CD <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/BridgeSupportBoat> .
_:GCSRCC3D48C26F7444A6965C0AA40EFEF7CD <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSRE83D8FB329364ADFAD11E6E474043A77 .
_:GCSRE83D8FB329364ADFAD11E6E474043A77 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Ferry> .
_:GCSRE83D8FB329364ADFAD11E6E474043A77 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSR21E99650242C457483E7024979B0DA4A .
_:GCSR21E99650242C457483E7024979B0DA4A <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Hulk> .
_:GCSR21E99650242C457483E7024979B0DA4A <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSR377818AB062448C49561EB1BFB68F603 .
_:GCSR377818AB062448C49561EB1BFB68F603 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/NavalAuxiliaryShip> .
_:GCSR377818AB062448C49561EB1BFB68F603 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSR2667361A860D4CB28D3E33320555FA27 .
_:GCSR2667361A860D4CB28D3E33320555FA27 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/NavalCombatantCraft> .
_:GCSR2667361A860D4CB28D3E33320555FA27 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSR3C71B5EE355A478994D3238C54B6CF4A .
_:GCSR3C71B5EE355A478994D3238C54B6CF4A <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/NavalCombatantShip> .
_:GCSR3C71B5EE355A478994D3238C54B6CF4A <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSR762C29D8144A498F89541C98075AA80C .
_:GCSR762C29D8144A498F89541C98075AA80C <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/NavalSubmarine> .
_:GCSR762C29D8144A498F89541C98075AA80C <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSR68C6612EE0C14858852097168555978E .
_:GCSR68C6612EE0C14858852097168555978E <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/NavalSupportCraft> .
_:GCSR68C6612EE0C14858852097168555978E <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:GCSR2916E14C900D48B182E961FD598B02C1 .
_:GCSR2916E14C900D48B182E961FD598B02C1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
<http://api.nsgreg.nga.mil/taxonomy/ntax/9.0/Pontoon> .
_:GCSR2916E14C900D48B182E961FD598B02C1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/22-rdf-syntax-nil> .
```

Figure 8 – Sample of NTAX individual resource (Entity Class “Hulk”) in N-Triples encoding

Annex A – Conformance

(Normative)

A.1 *Introduction*

Conformance is the fulfilment of specified requirements.[32] Conformance with the NSG Taxonomy Standard and associated NTAX content shall be determined based on the tests specified in this Annex. Any product claiming conformance with this NSG Taxonomy Standard in its entirety shall pass all the requirements specified in the abstract test suite in Section A.2. Conformance for use of a subset of the NTAX and for use of the NTAX with an extension is addressed in Sections A.1.3 and A.1.4, respectively.

A general explanation of conformance testing for the NSG Taxonomy is presented in this section, including relevant terminology. The conformance testing framework specified in Section A.2 is based on ISO 19105:2000 *Geographic information – Conformance and testing*. The definition of an abstract test suite for conformance testing appears in ISO 19105, together with an explanation of the testing framework. The format for conformance clauses is specified in ISO 19105, Annex A.

A.1.1 Terms and Definitions

A special terminology is used to describe the conformance testing framework. Terms that are defined in ISO 19105:2000 have a number in parentheses referring to the clause of that standard in which the term is defined.

Table 8 – Terms and Definitions for Conformance Testing

Term	Definition
abstract test case (ATC)	A generalized test for a particular requirement. (3.1) NOTE: An abstract test case is a formal basis for deriving executable test cases. One or more test purposes are encapsulated in the abstract test case. An abstract test case is independent of both the implementation and the values. It should be complete in the sense that it is sufficient to enable a test verdict to be assigned unambiguously to each potentially observable test outcome (*i.e.*, sequence of test events).
abstract test module (ATM)	A set of related **abstract test cases**. (3.3) NOTE: Abstract test modules may be nested in a hierarchical way.
abstract test suite (ATS)	An **abstract test module** specifying all the requirements to be satisfied for conformance. (3.4)
basic test	An initial **capability test** intended to identify clear cases of non-conformance. (3.6) NOTE: Basic tests may be used to determine whether to conduct further tests.
capability test	A test designed to determine whether an **implementation** under test conforms to a particular characteristic of a standard as described in the test purpose. (3.7) NOTE: Capability tests check that the capabilities claimed in an implementation conformance statement (ICS) are consistent with the observable capabilities of the implementation under test.
conformance	The fulfilment of specified requirements. (3.8) NOTE: Conformance may be claimed for data or software or services or for specifications including any profile or functional standard.
conformance testing	The testing of a product to determine the extent to which the product is a conforming **implementation**. (3.11)

[32] ISO 19105:2000 *Geographic information – Conformance and testing*.

contains	Includes a representation of the content of the NSG Taxonomy. NOTE1: A product may contain the NSG Taxonomy either directly (for example, by importing the NSG Taxonomy in one of its two official encodings), or by reference (for example, by accessing NSG Taxonomy entity classes as individual resources through the REST API component of the NSG Standards Registry). NOTE2: The representation included may be as a whole or in part, where use of a subset of the NSG Taxonomy is permitted.
implementation	A realization of a specification. (3.18) NOTE: In the context of the ISO geographic information standards, this includes specifications of geographic information services and datasets.
implementation conformance statement (ICS)	A statement of the options which have been implemented. (3.19) NOTE: This will allow the implementation to be tested for conformance against the relevant requirements, and against those requirements only. This statement shall contain only options within the framework of requirements specified in the relevant geographic information standards.
product	Data or software or a service. (3.8)
verdict	The result of a test. (6.4.4) NOTE: The value of a test verdict is one of: *pass*, *fail*, or *inconclusive*. Verdict criteria are specified by an abstract test case.

A.1.2 Conformance Testing Methodology

Conformance testing for the NSG Taxonomy is specified by this abstract test suite (ATS).

An ATS comprises all the abstract test cases needed to produce an overall verdict about the conformance of a candidate product. Abstract test cases may be collected in a set of related tests called an abstract test module. Abstract test modules may be nested. An abstract test suite includes test modules and other test cases arranged in a hierarchy of conformance tests.

Each abstract test case is designed to test a candidate product for conformance to a specific requirement. A test case has several components:

a) A test-case identifier;

b) A stated test purpose that is a precise description of the test objective and also indicates whether the requirement being tested is mandatory, conditional, or optional;

c) A description of the test method, specifying the test criteria that shall be used to determine the test verdict. A test may evaluate a multi-part requirement. The method indicates the way in which the test shall be conducted (*e.g.*, manual or automated). The test method may reference other clauses in the test suite.

d) References to one or more sections in the standard that identify the requirements addressed by the test.

e) The test type (either a basic test or a capability test).

Mandatory requirements are those which shall be observed in all cases. Conditional requirements shall be observed if the conditions set out in the specification apply. Optional requirements may be selected to suit the implementation, provided that any requirements applicable to the option are observed.[33]

In addition to an ATS, testing requires an implementation conformance statement (ICS) that declares which capabilities have been implemented for the product. This is especially important when there are options that may be implemented (or not), in order to evaluate the conformance of a particular implementation against the relevant requirements.

There are three primary options for conformance testing for the NSG Taxonomy:

1. Use of the complete NSG Taxonomy.

[33] ISO 19105:2000, Section 5.3.

2. Use of an allowable subset of the NSG Taxonomy.
3. Use of the NSG Taxonomy with allowable local extensions.

Products that claim conformance with the NSG Taxonomy shall support the official RDF/XML encoding of the NSG Taxonomy. Support for the N-Triples encoding is optional; however, if N-Triples are implemented, then it is mandatory to support the official N-Triples encoding.

The ATS for the NSG Taxonomy (Section A.2) specifies conformance evaluation of a product using the NSG Taxonomy as a whole, that is, with all of the content specified in the (mandatory) RDF/XML encoding and the (optional) N-Triples encoding. This annex also specifies the conditions under which selective use of the NSG Taxonomy conforms to the NTAX Standard and the conditions under which the NSG Taxonomy may be extended locally. The guidelines in Section A.1.3 and Section A.1.4 shall apply to modify the ATS for use in cases (2) and (3) above, respectively.

An implementation conformance statement (ICS) for a product to be tested for conformance to the NTAX Standard shall contain the following information regarding the capabilities that have been implemented for the product:

I. Identification of the NTAX content baseline (and thus the NTAX Standard edition) to which the product conforms.
II. Statement of whether the product optionally conforms to the N-Triples encoding. (Note: Conformance to the RDF/XML encoding is mandatory.)
III. Statement of whether the product uses NTAX content via active IRI-based Web links (using the REST API component of the NSG Standards Registry) or from a locally installed copy(ies) of the officially published technical artifacts.
IV. Statement of whether the product conforms to (a) the NTAX content in full, (b) a subset of the NTAX content, and/or (b) NTAX content (whether full or subset) with one or more local extension(s).
V. If the product conforms to a subset of the NTAX, then the ICS shall specify that subset for testing purposes, either via an active IRI-based Web link or as a loadable file.
VI. If the product conforms to the NTAX used with an extension(s), then the ICS shall specify that extension for testing purposes, either via an active IRI-based Web link or as a loadable file.
VII. Explanation of how to acquire authorized access to the system(s) where the product is installed, if needed to test the product.
VIII. Statement of whether the product categorizes (*i.e.*, classifies), individuals as instances of NTAX Entity Classes. If so, include with the ICS a representative sample of individual declarations for testing.

Abstract test cases may be automated for performance by a software system. Manual testing may be necessary when human judgment is required or when automated testing is too complex.

A.1.3 Testing Conformance for a Subset of the NSG Taxonomy

It is permissible to use only a part of the NTAX content if certain conditions are met. The selected set of NTAX components used is referred to here as the “Taxonomy Subset”. The conditions that must be met by the Taxonomy Subset are those of the ATS (Section A.2) with the following modifications:

1. For each NTAX entity class included in the Taxonomy Subset, all documentation properties for that entity class shall be included in the Taxonomy Subset, with their NSG Taxonomy values.
2. For each NTAX entity class included in the Taxonomy Subset, the Taxonomy Subset shall also include the generalization relationship and the superclass of that entity class as in the NSG Taxonomy.
3. For each NTAX entity class included in the Taxonomy Subset, the Taxonomy Subset shall also include all the entity classes that are siblings of that entity class in the NSG Taxonomy.
4. For each NTAX entity class included in the Taxonomy Subset, the Taxonomy Subset shall also include the DisjointClasses axiom from the NSG Taxonomy that includes that entity class (and its siblings).
5. For each NTAX entity class not included in the Taxonomy Subset, the Taxonomy Subset shall also exclude all the specializations (*i.e.*, subclasses) of that entity class.

A.1.4 Testing Conformance for the NSG Taxonomy with Extensions

A specific system or application may have requirements that make it desirable to specify new OWL classes that are subclasses of NTAX EntityClasses. For example, there may be a business need to categorize data using additional distinctions that are not made at a leaf level of the NSG Taxonomy.[34]

It is permissible to use the NSG Taxonomy with local extensions if certain conditions are met. The set of local extensions used with the NTAX is referred to here as the "Taxonomy Extension". In the requirements statements, the phrase "included in" means "belongs to the namespace of". The conditions that must be met by the Taxonomy Extension are:

1. The Taxonomy Extension shall be assigned a URI base different from that of the NSG Taxonomy, thus defining a separate namespace.
2. The Taxonomy Extension shall import either the whole NTAX content or a valid Taxonomy Subset of the NTAX (Section A.1.3). The imported taxonomy must pass either the ATS for the NSG Taxonomy (Section A.2) or for a valid subset of the NSG Taxonomy (Section A.1.3).
3. Components of the Taxonomy Extension shall not be included in the NTAX namespace.
4. Except for the use of a different URI base, the Taxonomy Extension shall conform to the NTAX information model that is specified in Section 5.2.3.
5. Documentation properties for each entity class in the Taxonomy Extension shall be populated as required by the NTAX information model.
6. Each entity class in the Taxonomy Extension shall have either a direct generalization (`rdfs:subClassOf`) relationship to an entity class in the NTAX, or an indirect generalization relationship to an entity class in the NTAX that is inferable through its direct generalization relationship to an entity class that is included the Taxonomy Extension.
7. If an entity class in the Taxonomy Extension is added to an intermediate node of the NTAX, then a DisjointClasses axiom shall be included to declare that the new entity class is pairwise disjoint with all of its sibling entity classes in the NTAX.
8. The Taxonomy Extension should include a DisjointClasses axiom for each set of two or more entity classes that are sibling subclasses of entity classes in the Taxonomy Extension.
9. If an entity class in the Taxonomy Extension has a direct generalization relationship to a leaf node of the NTAX hierarchy and the Taxonomy Extension contains other subclasses of that NTAX leaf node, then the Taxonomy Extension should include a DisjointClasses axiom for all the subclasses of that NSG Taxonomy leaf node.

A.1.5 Logical Structure of the Abstract Test Suite for the NSG Taxonomy

The abstract test suite for the NTAX contains four top-level test modules, each of which contains multiple test modules and/or test cases. The structure of the test suite is depicted in Figure 9.

[34] Local extensions may be proposed for inclusion in a future version of the NTAX content using the procedures described in Section **Error! Reference source not found.** about the Governance Process.

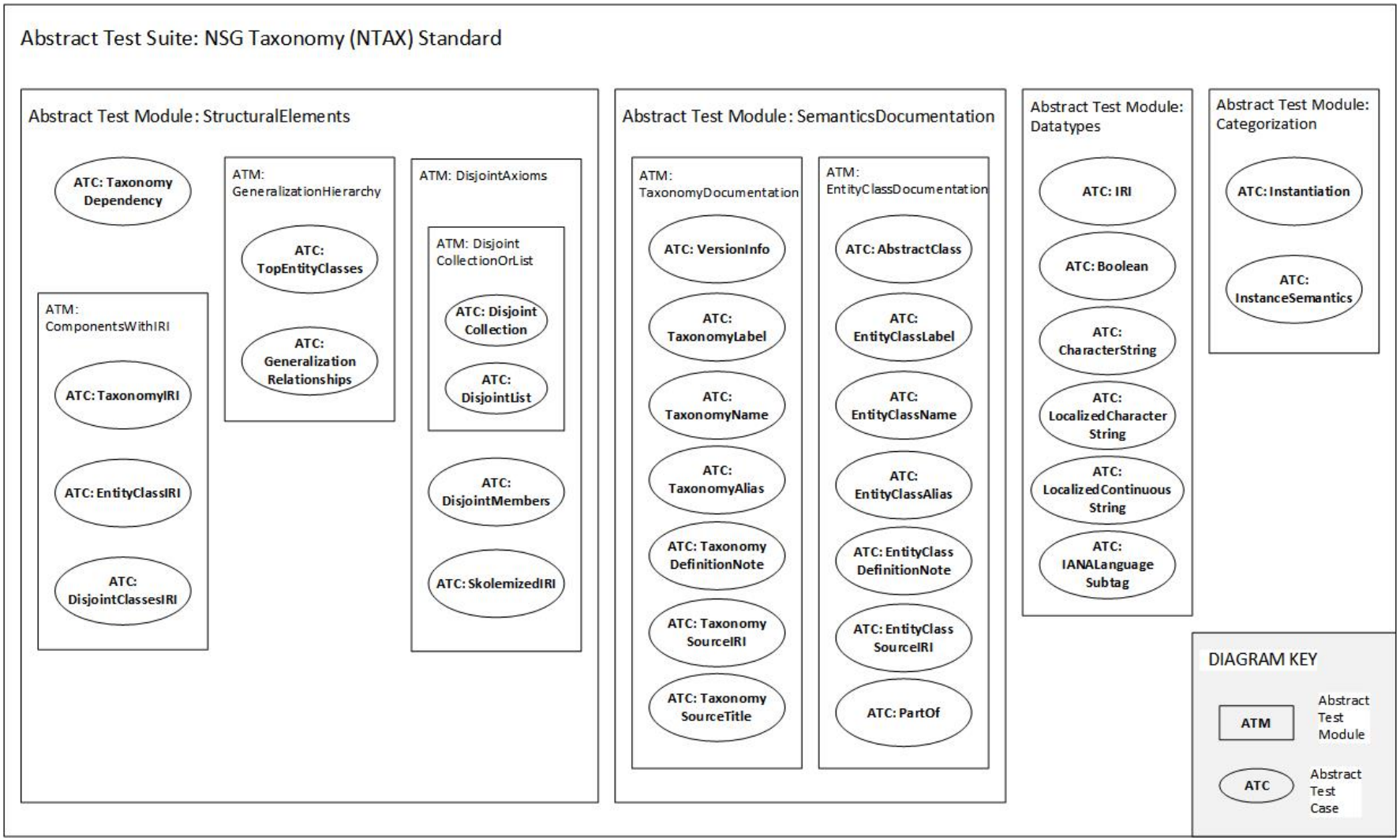

Figure 9 – Structure of Abstract Test Suite for NSG Taxonomy Standard

A.2 *Abstract Test Suite for the NSG Taxonomy (NTAX) Standard*

a) Test identifier: ATS_NTAX

b) Test purpose: Verify the conformance of the product with the taxonomy specification in the NTAX Standard and associated NTAX content. (**Mandatory**)

c) Test method: Inspect the product to determine that it contains the required NTAX structural elements (A.2.1), semantics documentation (A.2.2), and datatypes (A.2.3), in accordance with the requirements in the taxonomy specification, and that any instances declared in the product are consistent with the NTAX content (A.2.4).

d) Reference: NSG Taxonomy (NTAX) Standard, Section 5.

e) Test type: Basic

NOTE: If an information construct from the NTAX Standard is employed within a product, then the meaning and structure of that construct shall be preserved, and information regarding the corresponding construct shall be exactly as specified in the NTAX Standard. If NTAX content is employed within a product, then the meaning and structure of the content shall be consistent with the NSG Taxonomy in one of its official encodings.

A.2.1 Test Module for Conformance with Taxonomy Structure

a) Test identifier: StructuralElements

b) Test purpose: Verify the conformance of the product with the required NTAX structural elements. (**Mandatory**)

c) Test method: Inspect the product to determine that it contains the required structural elements, including all taxonomy dependencies (A.2.1.1), elements with identity (A.2.1.2), generalization hierarchy (A.2.1.3), and disjoint-classes axioms (A.2.1.4).

d) Reference: NTAX Standard (Table 4; Section 5.3.4, Section 5.4.2, Section 5.4.3, and Section 5.4.5)

e) Test type: Basic

A.2.1.1 Test Case for Taxonomy Dependency(ies)

a) Test identifier: TaxonomyDependency

b) Test purpose: Verify the conformance of the product with dependencies in the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product to determine that it contains all of the `owl:imports` that are declared in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.4.3).

e) Test type: Basic

A.2.1.2 Test Module for Components with Identity (IRIs)

a) Test identifier: ComponentsWithIRI

b) Test purpose: Verify the conformance of the product with all of the specified NTAX content having identity indicated by IRIs in the NTAX namespace. (**Mandatory**)

c) Test method: Inspect the product to determine that it contains all of the NTAX content having IRI values in the NTAX namespace, including the Taxonomy itself (A.2.1.2.1), EntityClasses (A.2.1.2.2), and (if applicable) DisjointClasses (A.2.1.2.3).

d) Reference: NTAX Standard (Table 4; Section 5.4.2, Section 5.4.3, and Section 5.4.5).

e) Test type: Basic

A.2.1.2.1 Test Case for Taxonomy with IRI

a) Test identifier: TaxonomyIRI

b) Test purpose: Verify the conformance of the product with the Taxonomy declaration and IRI value required for the indicated content baseline of the NSG Taxonomy. (**Mandatory**)

c) Test method: (1) Inspect the product in order to determine that it contains a declaration of the NSG Taxonomy as an `owl:Ontology`, in which the value in `rdf:about` is the IRI of the content baseline of the NSG Taxonomy indicated in the implementation conformance statement (ICS). (2) Inspect the product in order to determine that it contains an `owl:versionIRI` declaration in which the value is the IRI of the content baseline of the NSG Taxonomy indicated in the ICS. The two values for the content baseline IRI shall be identical. The IRI for the NSG Taxonomy shall always identify the applicable content baseline by utilizing an IRI that indicates the NSG Taxonomy version as identified in the ICS.

d) Reference: NTAX Standard (Table 4; Section 5.4.2, Section 5.4.3)

e) Test type: Basic

A.2.1.2.2 Test Case for Entity Classes with IRIs

a) Test identifier: EntityClassIRI

b) Test purpose: Verify the conformance of the product with the EntityClass declarations and IRI values required for EntityClasses in the indicated content baseline of the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine (1) that it contains an `owl:Class` declaration for each EntityClass in the content baseline of the NSG Taxonomy indicated in the implementation conformance statement (ICS), and (2) that the value in `rdf:about` for each EntityClass is the value of the classIRI in the NSG Taxonomy. The IRI for an NSG Taxonomy EntityClass shall always identify the applicable content baseline by utilizing an IRI that indicates the NSG Taxonomy version as identified in the ICS.

d) Reference: NTAX Standard (Table 4; Section 5.4.2 and Section 5.4.3)

e) Test type: Basic

A.2.1.2.3 Test Case for DisjointClasses with IRIs

a) Test identifier: DisjointClassesIRI

b) Test purpose: Verify the conformance of the product with the declaration of DisjointClasses using Skolemized IRIs. (*Conditional* on the implementation being in the N-Triples encoding)

c) Test method: See A.2.1.4.3.

d) Reference: NTAX Standard (Section 5.4.5.3).

e) Test type: Capability test

A.2.1.3 Test Module for Generalization (Subclass) Hierarchy

a) Test identifier: GeneralizationHierarchy

b) Test purpose: Verify the conformance of the product with the complete Taxonomy generalization hierarchy (*i.e.*, subclass tree) of the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains all of the most general concepts (A.2.1.3.1) in the NSG Taxonomy and all of the `rdfs:subClassOf` relationships (A.2.1.3.2) between EntityClasses in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.5 and Section 5.4.3)

e) Test type: Basic

A.2.1.3.1 Test Case for Top Entity Classes

a) Test identifier: TopEntityClasses

b) Test purpose: Verify the conformance of the product with the most basic (*i.e.*, top) EntityClasses in the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains all of the most basic EntityClasses in the NSG Taxonomy, *i.e.*, the EntityClasses that are not subclasses of any other EntityClass in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.5 and Section 5.4.3)

e) Test type: Basic

A.2.1.3.2 Test Case for Generalization Relationships

a) Test identifier: GeneralizationRelationships

b) Test purpose: Verify the conformance of the product with all the generalization relationships in NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains all of the `rdfs:subClassOf` relationships that are declared between EntityClasses in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.5 and Section 5.4.3)

e) Test type: Basic

A.2.1.4 Test Module for Disjointness Axioms

a) Test identifier: DisjointAxioms

b) Test purpose: Verify the conformance of the product with the requirement to represent the disjointness constraints on all sets of sibling EntityClasses in the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required DisjointClasses axioms (A.2.1.4.1) used for declaring the pairwise disjointness of sibling EntityClasses (A.2.1.4.2), with Skolemized IRIs (A.2.1.4.3) where required.

d) Reference: NTAX Standard (Section 5.4.3, Section 5.4.5.2, and Section 5.4.5.3)

e) Test type: Basic

NOTE: Sibling EntityClasses are those which have the same EntityClass as their generalization (*i.e.*, superclass).

A.2.1.4.1 Test Module for Disjoint Collection or List

a) Test identifier: DisjointCollectionOrList

b) Test purpose: Verify the conformance of the product with the requirement to include components to represent DisjointClasses axioms for all sets of sibling EntityClasses. (**Mandatory**)

c) Test method: Inspect the product in order to determine that (1) it contains the required structural components appropriate to the encoding (A.2.1.4.1.1 or A.2.1.4.1.2) for stating the DisjointClasses axioms for sibling subclasses, and (2) it contains a DisjointClasses component for every set of sibling EntityClasses in the NSG Taxonomy.

d) Reference: NTAX Standard (Section 5.4.3, Section 5.4.5.2, and Section 5.4.5.3)

e) Test type: Basic

A.2.1.4.1.1 Test Case for DisjointClasses Collection

a) Test identifier: DisjointCollection

b) Test purpose: Verify the conformance of the product with the requirement to represent DisjointClasses as collections of EntityClasses. (*Conditional* on the product being in the RDF/XML encoding; Note: conformance to the RDF/XML encoding is **Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required `owl:AllDisjointClasses` component, declared as a (RDF parseType) collection, for each set of sibling EntityClasses in the NSG Taxonomy.

d) Reference: NTAX Standard (Section 5.4.3 and Section 5.4.5.2)

e) Test type: Capability test

A.2.1.4.1.2 Test Case for DisjointClasses List

a) Test identifier: DisjointList

b) Test purpose: Verify the conformance of the product with the requirement to represent DisjointClasses as lists of EntityClasses. (*Conditional* on the product being in the N-Triples encoding)

c) Test method: Inspect the product in order to determine that it contains the required http://www.w3.org/2002/07/owl#AllDisjointClasses component, structured as an RDF List, for each set of sibling EntityClasses in the NSG Taxonomy.

d) Reference: NTAX Standard (Section 5.4.3 and Section 5.4.5.3)

e) Test type: Capability test

A.2.1.4.2 Test Case for Members of DisjointClasses

a) Test identifier: DisjointMembers

b) Test purpose: Verify the conformance of the product with the required enumeration of all sibling EntityClasses within a DisjointClasses axiom. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains (1) [for the RDF/XML encoding] a complete collection of the `owl:members` of each `owl:AllDisjointClasses` component, or (2) [for the N-Triples encoding] a complete list of the http://www.w3.org/2002/07/owl#members of each http://www.w3.org/2002/07/owl#AllDisjointClasses component.

d) Reference: NTAX Standard (Section 5.4.5.2 and Section 5.4.5.3)

e) Test type: Capability test

A.2.1.4.3 Test Case for Skolemized IRIs

a) Test identifier: SkolemizedIRI

b) Test purpose: Verify the conformance of the product with the use of a Skolemized IRI required as the identifier for each representation of a DisjointClasses axiom. (*Conditional* on the product being in the N-Triples encoding)

c) Test method: Inspect the product in order to determine that it contains a Skolemized IRI as the identifier for each instance of http://www.w3.org/2002/07/owl#AllDisjointClasses in the N-Triples encoding.

d) Reference: NTAX Standard (Section 5.4.5.3)

e) Test type: Capability

NOTE: The URI base for a Skolemized IRI differs from the URI base of the NSG Taxonomy (*e.g.*, NTAX URI base for the November 2016 content baseline: http://api.nsgreg.nga.mil/taxonomy/ntax/base17Feb/...). Example of a Skolemized IRI: http://api.nsgreg.nga.mil/.well-known/genid/C26B41F050384C878C00D7D462C2730F.

A.2.2 Test Module for Documentation of Semantics

a) Test identifier: SemanticsDocumentation

b) Test purpose: Verify the conformance of the product with the documentation properties for the NSG Taxonomy and its component EntityClasses as specified in the taxonomy specification in the NTAX Standard. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains all of the required documentation properties with the correct values for the NSG Taxonomy (A.2.2.1) and its component EntityClasses (A.2.2.2).

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.1 Test Module for Taxonomy Documentation

a) Test identifier: TaxonomyDocumentation

b) Test purpose: Verify the conformance of the product with the documentation properties required for the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required ontology documentation properties with the correct values (A.2.2.1.1, A.2.2.1.2, A.2.2.1.3, A.2.2.1.4, A.2.2.1.5, A.2.2.1.6, and A.2.2.1.7) for the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.1.1 Test Case for Taxonomy Version Information

a) Test identifier: VersionInfo

b) Test purpose: Verify the conformance of the product with documentation of the required version information for the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required documentation property `owl:versionInfo` with the value specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.1.2 Test Case for Taxonomy Label

a) Test identifier: TaxonomyLabel

b) Test purpose: Verify the conformance of the product with the documentation of the label for the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required documentation `rdfs:label` with the value as specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.1.3 Test Case for Taxonomy Name

a) Test identifier: TaxonomyName

b) Test purpose: Verify the conformance of the product with documentation of the name for the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required documentation property `skos:prefLabel` with the value as specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.1.4 Test Case for Taxonomy Alias

a) Test identifier: TaxonomyAlias

b) Test purpose: Verify the conformance of the product with documentation of alias(es) for the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required documentation property `skos:altLabel` with the value “NTAX” as specified for the NSG Taxonomy.

d) Reference: NTAX Standard (Table 5; Section 5.4.3)

e) Test type: Basic

A.2.2.1.5 Test Case for Taxonomy Definition Note

a) Test identifier: TaxonomyDefinitionNote

b) Test purpose: Verify the conformance of the product with documentation of the required definitionNote for the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required documentation property `skos:definition` with the value as specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.1.6 Test Case for Taxonomy Source Reference

a) Test identifier: TaxonomySourceIRI

b) Test purpose: Verify the conformance of the product with documentation of the required sourceIRI for the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required documentation property `rdfs:isDefinedBy` with the value as specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.1.7 Test Case for Taxonomy Source Title

a) Test identifier: TaxonomySourceTitle

b) Test purpose: Verify the conformance of the product with the documentation of the title of the Standard document on which the NSG Taxonomy is based. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required documentation property `dct:source` with the value as specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.2 Test Module for EntityClass Documentation

a) Test identifier: EntityClassDocumentation

b) Test purpose: Verify the conformance of the product with the required documentation properties for the EntityClasses in the NSG Taxonomy. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains the required EntityClass documentation properties with the correct values (A.2.2.2.1, A.2.2.2.2, A.2.2.2.3, A.2.2.2.4, A.2.2.2.5, A.2.2.2.6, and A.2.2.2.7) for each EntityClass in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.2.1 Test Case for Abstract Class

a) Test identifier: AbstractClass

b) Test purpose: Verify the conformance of the product with the indication of all abstract EntityClasses. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it contains an assertion of the Boolean documentation property `iso19150-2:isAbstract` with value TRUE for each abstract EntityClass in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

NOTE: Abstract classes shall not be directly instantiated. See the Test Case A.2.4.1.

A.2.2.2.2 Test Case for EntityClass Label

a) Test identifier: EntityClassLabel

b) Test purpose: Verify the conformance of the product with the documentation of the label for each EntityClass. (**Mandatory**)

c) Test method: Inspect the product in order to determine that each EntityClass has the required documentation property `rdfs:label` with the value as specified in the NSG Taxonomy. The value is the same as the terminal component of the EntityClassIRI.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.2.3 Test Case for EntityClass Name

a) Test identifier: EntityClassName

b) Test purpose: Verify the conformance of the product with documentation of the name for each EntityClass. (**Mandatory**)

c) Test method: Inspect the product in order to determine that each EntityClass has the required documentation property `skos:prefLabel` with the value for the preferred human-readable name of the class as specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

NOTE: The preferred human-readable name of the EntityClass should be used to refer to that class in navigation menus, browse trees, or other displays.

A.2.2.2.4 Test Case for EntityClass Alias

a) Test identifier: EntityClassAlias

b) Test purpose: Verify the conformance of the product with documentation of the optional alias(es), if any, for each EntityClass. (**Mandatory**)

c) Test method: Inspect the product in order to determine that each EntityClass has the documentation property `skos:altLabel` with the value(s) specified in the NSG Taxonomy, if there are any. Aliases are optional elements of the NSG Taxonomy representation.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.2.5 Test Case for EntityClass Definition Note

a) Test identifier: EntityClassDefinitionNote

b) Test purpose: Verify the conformance of the product with documentation of the required definitionNote for each EntityClass. (**Mandatory**)

c) Test method: Inspect the product in order to determine that each EntityClass has the required documentation property `skos:definition` with the value as specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3**Error! Reference source not found.**)

e) Test type: Basic

A.2.2.2.6 Test Case for Source Reference

a) Test identifier: EntityClassSourceIRI

b) Test purpose: Verify the conformance of the product with documentation of the required sourceIRI for each EntityClass. (**Mandatory**)

c) Test method: Inspect the product in order to determine that each EntityClass has the required documentation property `rdfs:isDefinedBy` with the value specified in the NSG Taxonomy.

d) Reference: NTAX Standard (Table 4; Section 5.3.6 and Section 5.4.3)

e) Test type: Basic

A.2.2.2.7 Test Case for Part-of

a) Test identifier: PartOf

b) Test purpose: Verify the conformance of the product with the required use of `dct:isPartOf` declarations. (*Conditional* on a product that represents the taxonomy EntityClasses in separate resource files)

c) Test method: Inspect the product in order to determine that in each individual resource file, the EntityClass has the required documentation property `dct:isPartOf` with the value of the NSG Taxonomy IRI.

d) Reference: NTAX Standard (Section 6.2.3)

e) Test type: Basic

A.2.3 Test Module for Datatype Conformance

a) Test identifier: Datatypes

b) Test purpose: Verify the conformance of the product with the datatypes specified in the taxonomy specification in the NTAX Standard. (**Mandatory**)

c) Test method: Inspect the product to determine that it uses the required datatypes and encodings for the IRI (A.2.3.1), Boolean (A.2.3.2), CharacterString (A.2.3.3), LocalizedCharacterString (A.2.3.4), LocalizedContinuousString (A.2.3.5), and IANALanguageSubtag (A.2.3.6) value types specified in the NTAX Standard taxonomy specification.

d) Reference: NTAX Standard (Table 4; Section 5.2.4 **Error! Reference source not found.**and Section 5.4.4)

e) Test type: Basic

A.2.3.1 Test Case for IRI Datatype

a) Test identifier: IRI

b) Test purpose: Verify the conformance of the product with the IRI datatype. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it uses values in the range of the required datatype `xsd:anyURI` for properties specified with the value type IRI in the NSG Taxonomy information model.

d) Reference: NTAX Standard (Table 4; Section 5.2.4.7, Section 5.4.2.5, and Section 5.4.4)

e) Test type: Basic

A.2.3.2 Test Case for Boolean Datatype

a) Test identifier: Boolean

b) Test purpose: Verify the conformance of the product with the Boolean datatype. (**Mandatory**)

c) Test method: Inspect the product in order to determine that it uses values in the range of the required datatype `xsd:boolean` for properties specified with the value type Boolean in the NSG Taxonomy information model.

d) Reference: NTAX Standard (Table 4; Section 5.2.4.6 and Section 5.4.4)

e) Test type: Basic

A.2.3.3 Test Case for CharacterString Datatype

a) Test identifier: CharacterString

b) Test purpose: Verify the conformance of the product with the CharacterString datatype. (**Mandatory**)

c) Test method: (1) Inspect the product in order to determine that it uses values in the range of the required datatype `rdf:PlainLiteral` for properties specified with the value type CharacterString in the NSG Taxonomy information model. (2) Values that are character strings (*e.g.*, values of `xsd:string`) with no language tag satisfy the requirements for the value type CharacterString in the NSG Taxonomy information model. (3) Optionally, a language tag may be present.

d) Reference: NTAX Standard (Table 4; Section 5.2.4.2**Error! Reference source not found.** and Section 5.4.4**Error! Reference source not found.**)

e) Test type: Basic

A.2.3.4 Test Case for LocalizedCharacterString Datatype

a) Test identifier: LocalizedCharacterString

b) Test purpose: Verify the conformance of the product with the LocalizedCharacterString datatype. (**Mandatory**)

c) Test method: (1) Inspect the product in order to determine that it uses values in the range of the required datatype `rdf:PlainLiteral` for properties specified with the value type LocalizedCharacterString in the NSG Taxonomy information model. (2) In order to satisfy the requirements for a LocalizedCharacterString in the NSG Taxonomy information model, a value must include both a character string and a language tag.

d) Reference: NTAX Standard (Table 4; Section 5.2.4.3**Error! Reference source not found.** and Section 5.4.4**Error! Reference source not found.**)

e) Test type: Basic

A.2.3.5 Test Case for LocalizedContinuousString Datatype

a) Test identifier: LocalizedContinuousString

b) Test purpose: Verify the conformance of the product with the LocalizedContinuousString datatype. (**Mandatory**)

c) Test method: (1) Inspect the product in order to determine that it uses values in the range of the required datatype `rdf:PlainLiteral` for properties specified with the value type LocalizedContinuousString in the NSG Taxonomy information model. (2) In order to satisfy the requirements for a LocalizedCharacterString in the NSG Taxonomy information model, a value must include both a character string and a language tag. (3) The string portion of the value must not contain any space characters (unless those are encoded using '%20').

d) Reference: NTAX Standard (Table 4; Section 5.2.4.4**Error! Reference source not found.** and Section 5.4.4**Error! Reference source not found.**)

e) Test type: Basic

A.2.3.6 Test Case for IANALanguageSubtag Datatype

a) Test identifier: IANALanguageSubtag

b) Test purpose: Verify the conformance of the product with the IANALanguageSubtag datatype. (**Mandatory**)

c) Test method: (1) Inspect the product to determine that it uses values in the range of the required language datatype for properties specified with the value types LocalizedCharacterString and LocalizedContinuousString in the NSG Taxonomy information model. (2) In order to satisfy the requirements for an IANALanguageSubtag in the NSG Taxonomy information model, a value must belong to the set of two-character, lowercase values specified in BCP 47).

d) Reference: NTAX Standard (Table 4; Section 5.2.4.5**Error! Reference source not found.** and Section 5.4.4**Error! Reference source not found.**)

Test type: Basic

A.2.4 Test Module for Categorization

a) Test identifier: Categorization

b) Test purpose: Verify the conformance of the product with the option to categorize individuals as instances of NSG Taxonomy classes consistent with the structure and semantics of the NSG Taxonomy. (*Conditional*)

c) Test method: Inspect the product to determine that instance declarations are made correctly (A.2.4.1) and the categorization is consistent with the semantics of the NSG Taxonomy (A.2.4.2).

d) Reference: OWL 2 Structural Specification and NSG Taxonomy content

e) Test type: Capability

A.2.4.1 Test Case for Declarations of Instances

a) Test identifier: Instantiation

b) Test purpose: Verify the conformance of the product with the option to categorize individuals as instances of NSG Taxonomy classes. (*Conditional*)

c) Test method: Inspect the product in order to determine that (1) for each individual declared to be an instance of an EntityClass belonging the NSG Taxonomy, the product contains (a) the required declaration of the individual as an `owl:NamedIndividual` and (b) the required `rdf:type` declaration linking that individual to the NSG Taxonomy EntityClass (using the IRI for the Entity Class); (2) no individual in the product is declared to be a direct instance of any NSG Taxonomy EntityClass having the property `iso19150-2:isAbstract` with the value ‘TRUE’; and (3) no individual in the product is assigned an IRI in the NSG Taxonomy namespace.

d) Reference: OWL 2 Structural Specification (Section 5.6); OWL 2 Mapping to RDF Graphs

e) Test type: Basic

A.2.4.2 Test Case for Semantics of Instances

a) Test identifier: InstanceSemantics

b) Test purpose: Verify the conformance of each individual declared to be an instance of an NSG Taxonomy EntityClass with the semantics of that class. (*Conditional*)

c) Test method: Inspect the product to determine that any individual declared to be an instance of an EntityClass in the NSG Taxonomy (1) conforms to the DefinitionNote of that EntityClass in the NSG Taxonomy; (2) conforms to the semantics (including the DefinitionNote) of every generalization of the EntityClass of which it is declared an instance; and (3) is not declared an instance of more than one sibling subclass in the NSG Taxonomy.

d) Reference: NSG Taxonomy Standard (Table 4; Section 5.3.5**Error! Reference source not found.** and Section 5.4.3); NSG Taxonomy content

e) Test type: Capability test

Annex B – Inspecting NSG Taxonomy (NTAX) Content

(Informative)

B.1 *Introduction*

The NSG Taxonomy is declared as an OWL ontology and may be viewed and edited in ontology tools that load OWL encoded in the mandatory RDF/XML syntax or the optional N-Triples format. Protégé is a widely used, free, open-source ontology editor that may be used for this purpose.[35]

This Annex illustrates the use of the open-source ontology viewer and editor Protégé for visualization and inspection of NTAX content.

B.2 *NTAX Content Inspection using the Protégé Ontology Tool*

B.2.1 Protégé: An Open-Source Ontology Tool for Viewing W3C OWL ontologies

The Protégé ontology editor, developed by Stanford University, is a free, open-source ontology development tool that may be used to view NTAX content. Protégé has a graphical user interface that displays the class hierarchy, with detail panes for examining individual concepts. Protégé also offers plug-ins for visualization, although the size of the NTAX makes visualization of the entire ontology difficult. Protégé is available for download online at http://protege.stanford.edu/. The web site also provides documentation and training materials.

NOTE: Protégé may attempt to find external resources referenced by the ontology being opened. If those are not accessible from the system where Protégé is installed, Protégé will prompt the user to resolve the issue. Two options are available: (1) Click “No” to have Protégé proceed to open the ontology file without accessing the external resources; (2) locate and copy the external resources and provide them in files accessible to Protégé within the installation system (for assistance, users should consult their system administrators).

B.2.2 Viewing NTAX Content using Protégé and its Plug-ins

The following screen captures show various ways of viewing information about the NSG Taxonomy as a whole, NTAX components, and their documentation metadata, using the open-source ontology tool, Protégé (version 5.1.0).

[35] Protégé may be downloaded online from http://protege.stanford.edu/.

Figure 10 shows the metadata information describing the NTAX ontology, as presented on the Active Ontology tab of the Protégé GUI.

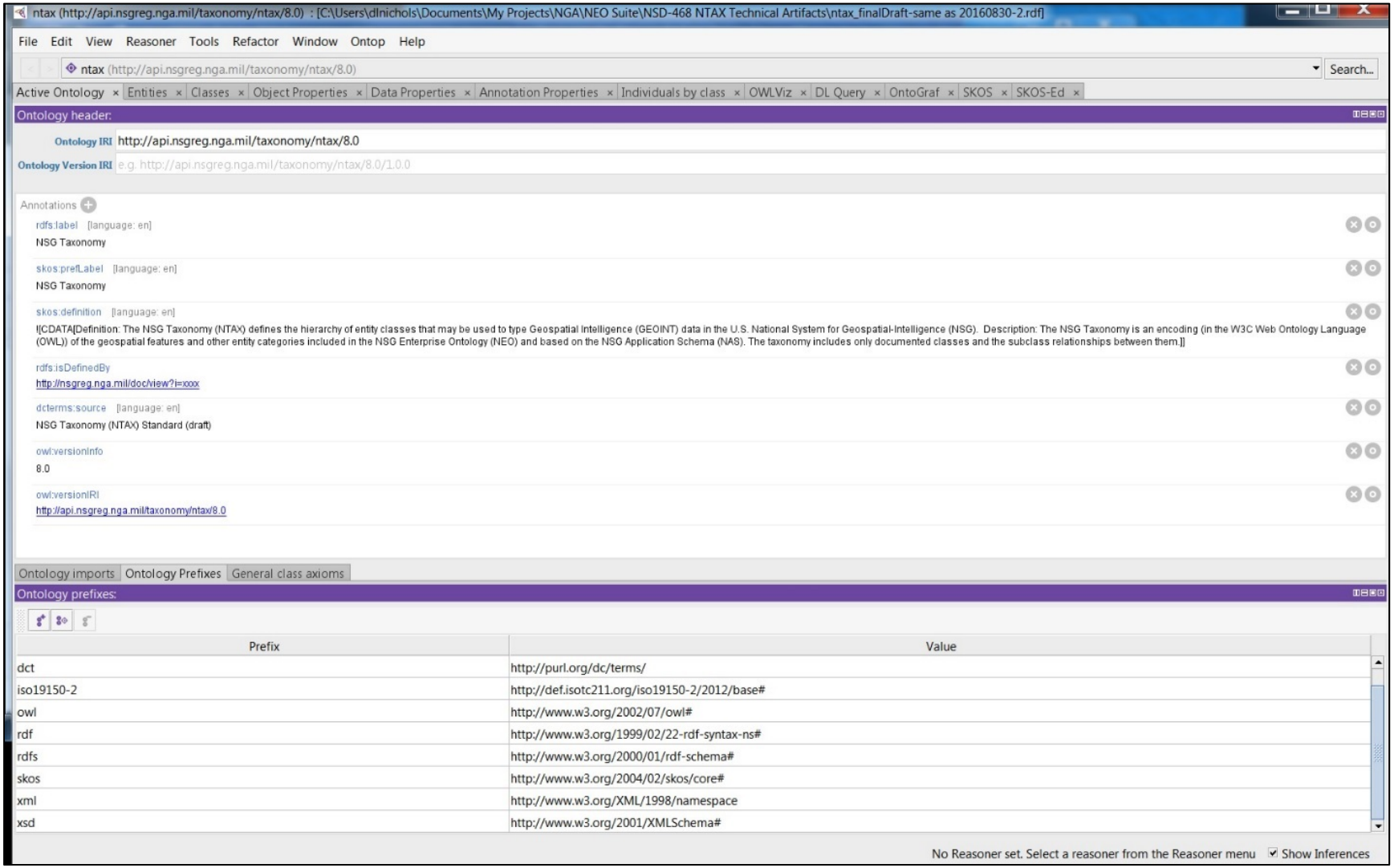

Figure 10 – NTAX Described on the Active Ontology Tab of the Protégé Tool

Figure 11 shows the NTAX class hierarchy as presented in the Protégé GUI (left pane), with a detailed display (upper right pane) of the documentary information for the highlighted class “FeatureEntity”, including: the preferred name of the concept, its definition, and a link to the source (*i.e.*, the NAS entry) in which the concept is defined.

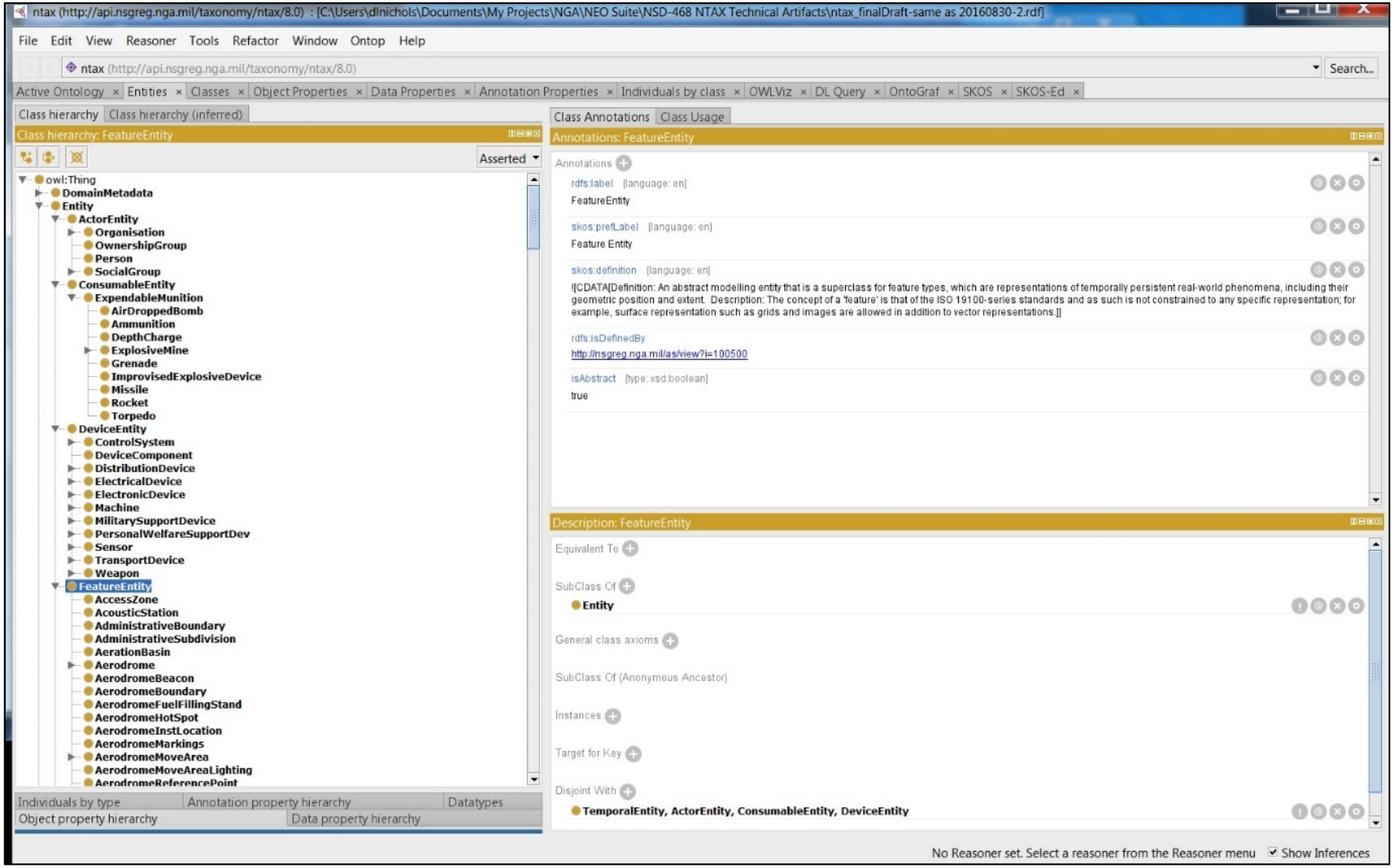

Figure 11 – Protégé View of NTAX Hierarchy with Individual Concept (Feature Entity)

Figure 12 shows a graphical representation of the subclass hierarchy under the NTAX class ActorEntity, using the Protégé plug-in OntoGraf. Class documentation is displayed with each node. It is worth noting that the conventions of the OntoGraf diagram differ from those of UML (*e.g.*, in the position and direction of the arrows on the generalization relation).

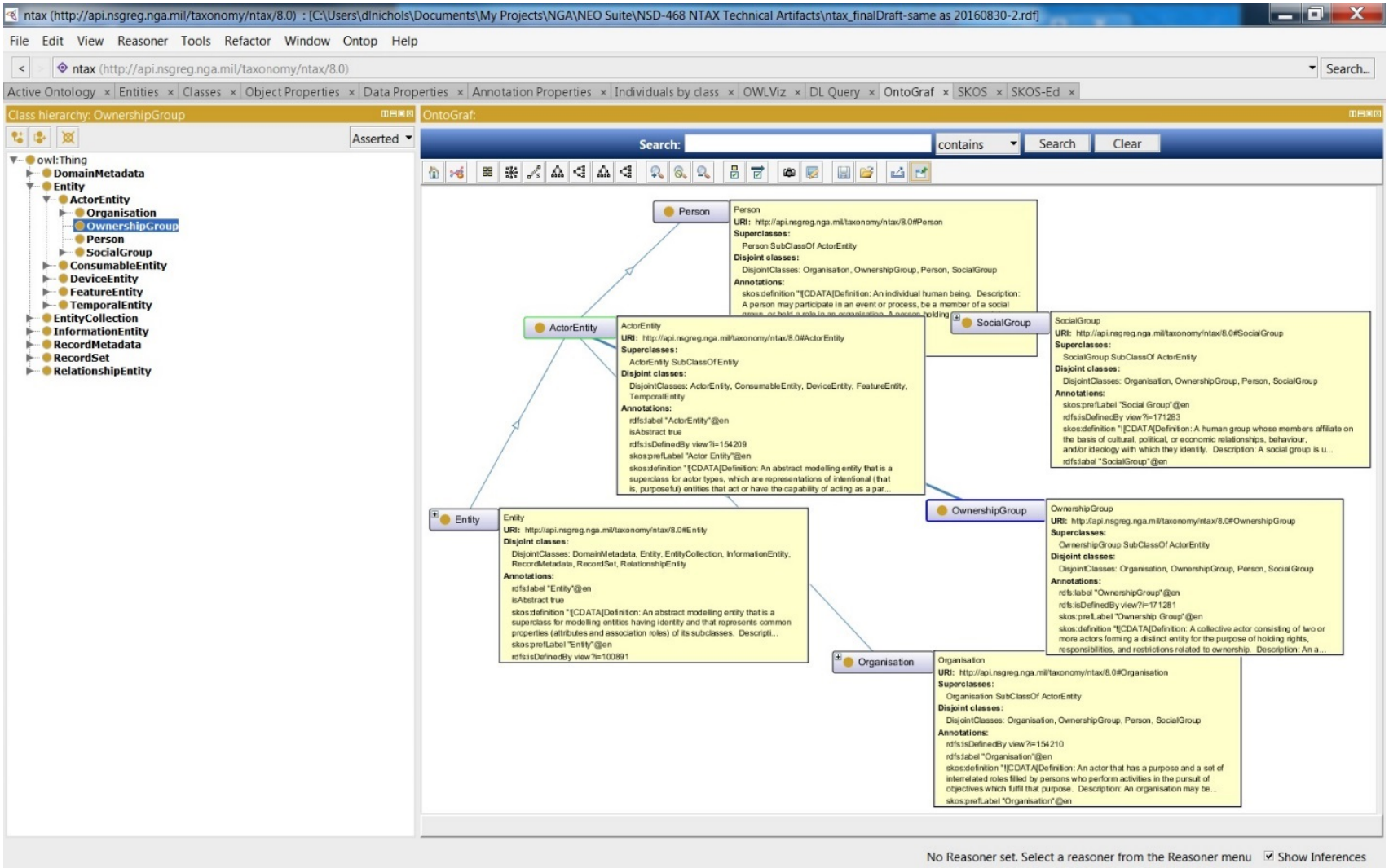

Figure 12 – OntoGraf Plug-in View of NTAX Actor Entity Hierarchy with Documentation

An alternative graphical display is presented in Figure 13, which shows the subclasses of Building using the Protégé plug-in called OWLViz.

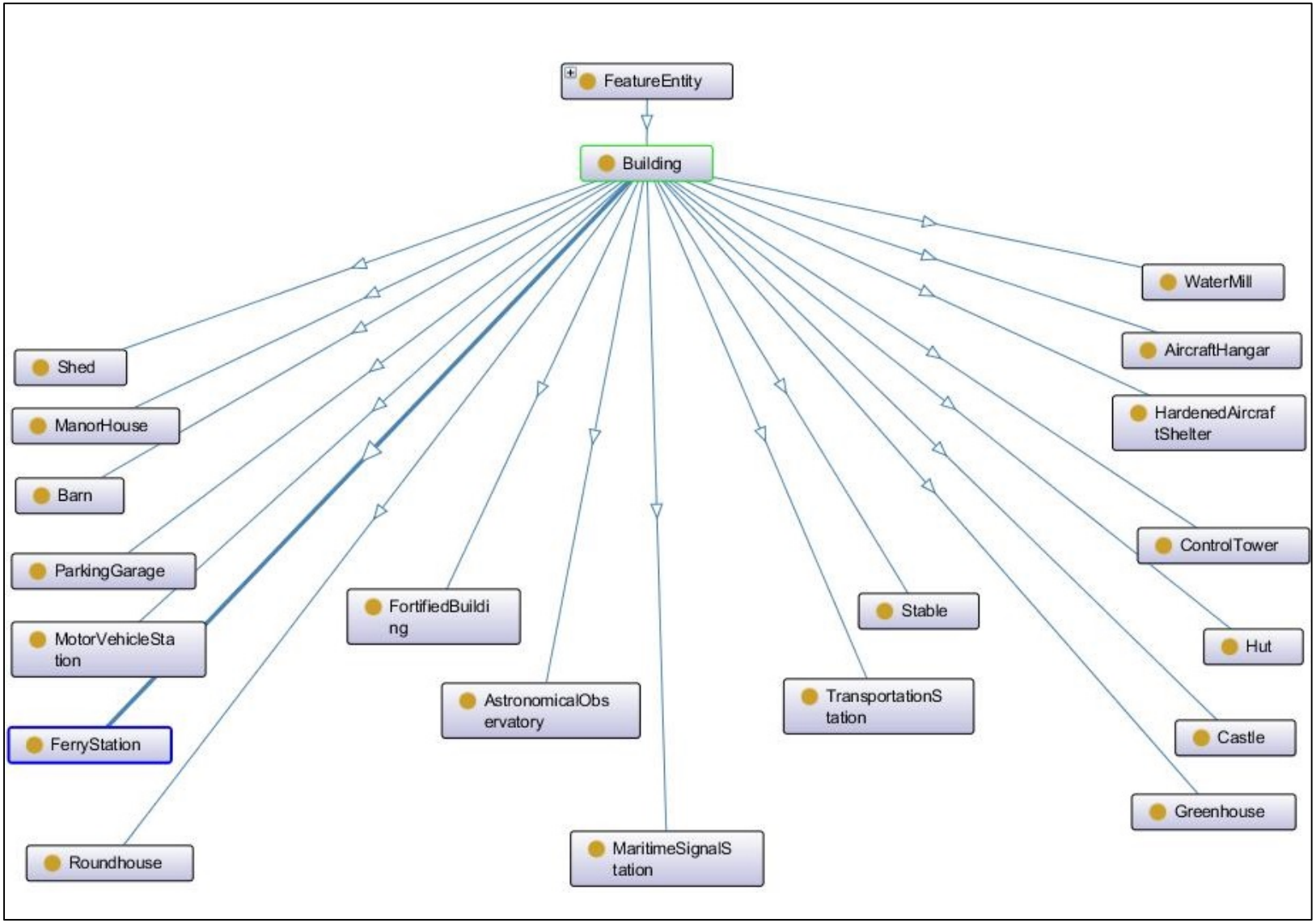

Figure 13 – OWLViz Plug-in View of Subclasses of NTAX Building

Annex C – Derivation of the NSG Taxonomy

(Informative)

C.1 *Introduction*

The NSG Application Schema (NAS) is the logical model for geospatial data in the National System for Geospatial Intelligence (NSG). The purpose of the NAS is to ensure a common understanding of GEOINT data across the enterprise.

> Semantic conformance [to the NAS] ensures that GEOINT-enabled missions "mean the same thing" at all times/places, regardless of whether their underlying information systems are involved in electronic information exchange. This also ensures that GEOINT-aware warfighters share and employ the same understanding of GEOINT semantics regardless of the mission-context in which that GEOINT is employed. (NAS – Part 1)

The information content of the NSG Taxonomy is derived from the information content of the NSG Application Schema (NAS) through an automated, rule-based process based on ISO 19150-2:2015 *Geographic information – Ontology – Part 2: Rules for developing ontologies in the Web Ontology Language (OWL)*.

This Annex describes the structural requirements for the NSG Taxonomy and specifies the derivation of the content of NSG Taxonomy encoding elements from the information content of the NAS.

C.2 *NTAX Structural Requirements*

The NTAX Standard defines a structural specification for the NSG Taxonomy that includes an information model based on the generalization hierarchy of entity classes in the ISO 19109 General Feature Model (GFM) and an encoding based on the W3C Web Ontology Language (OWL 2).

The NAS conceptual metamodel conforms to the General Feature Model (GFM) as specified in *ISO 19109, Geographic information – Rules for application schema*. The NAS model extends the ISO 19109 GFM to add human-readable semantics to the application schema, as defined in NAS – Part 1.

The NTAX information model is also based on the ISO 19109 GFM and incorporates additional human-readable semantics, as described in Section 5.2 of this Standard. However, while the NAS is a complete application schema for defining NSG-specific data sets, the NTAX information model represents only the entity concepts and their hierarchical class structure. Non-hierarchical relationships and entity properties are not represented in the NTAX information model.

A taxonomy is typically visualized as a tree structure. In one perspective, the tree is depicted as branching upward, with the most general concept(s) at the bottom and the more specialized concepts higher in the tree. An alternative view depicts the tree as inverted, with the most general concept(s) at the top and the more specialized concepts spreading out below. In the upright-tree approach, the most general concept(s) are called "root(s)", while in the inverted-tree approach, the most general concept(s) are called "top(s)". The NTAX Standard uses the inverted-tree perspective and calls the most general concepts "tops".

Taxonomies may have multiple tops.[36] The number of tops depends on the number of basic concepts in the domain described by the taxonomy. Each basic concept is represented by a class that is the top of its own taxonomic tree.

The NSG Taxonomy is a class-generalization hierarchy with multiple top classes rather than a single top. The NTAX contains multiple top classes because the taxonomy domain includes high-level concepts for which there is no generalization.[37]

At the present time, there are seven top classes in the NTAX (listed by their human-readable names):

1. Entity

[36] In the upright-tree perspective, this would be "multi-rooted".

[37] "Every individual in the OWL-encoded universe of discourse is a member of the class `owl:Thing`. Thus each user-defined class is implicitly a subclass of `owl:Thing`." (OWL Web Ontology Language Guide. Available online at: http://www.w3.org/TR/2004/REC-owl-guide-20040210: Section 3.1.1) Consequently, when viewed in an ontology viewer such as Protégé, the seven top classes of the NTAX hierarchy will all be subsumed under the most general class in the OWL vocabulary, `owl:Thing`.

2. Entity Collection
3. Information Entity
4. Relationship Entity
5. Domain Metadata
6. Record Set
7. Record Metadata

The first five of the top NTAX Entity Classes listed above are derived directly from conceptually equivalent classes in the NAS. The class 'Entity' contains subclasses for the main types of objects in most domains, including: actors, devices, features, and temporal entities (such as events). 'Entity Collection' represents collections of objects, which are specialized by their object type. 'Information Entity' contains subclasses representing different types of intangible phenomena in a domain. 'Relationship Entity' contains subclasses for types of relationships between objects. 'Domain Metadata' contains subclasses for types of information about evaluated properties used to describe domain entities.

The last two top classes listed above ('Record Set' and 'Record Metadata') are not directly derived from conceptually equivalent classes in the NAS. Instead, these two are defined in the NSG Taxonomy as superclasses to collect two groups of more specific NTAX Entity Classes directly derived from NAS classes that have no superclass in the NAS.[38] 'Record Set' has subclasses that represent different kinds of collections of resources, while 'Record Metadata' has subclasses that represent different kinds of metadata about recorded information. The NTAX Entity Classes included in 'Record Metadata' are derived directly from NAS Entity Type Classes that in the NAS are subclasses of classes external to the NAS namespace. Thus, the top classes 'RecordSet' and 'Record Metadata' enable categorization of similar concepts that would otherwise not be found together within the 'ntax' namespace.

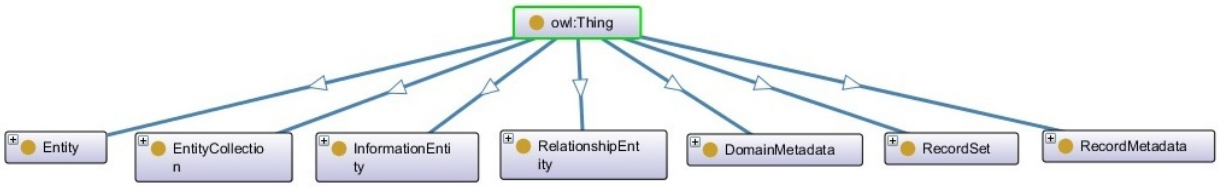

Figure 14 – Top Entity Classes of the NSG Taxonomy[39]

The set of top classes in the NTAX, including those that are not derived directly from the NAS, may change in the future based on requirements approved through the NTAX governance process (Section 6).

C.3 *NTAX Encoding Element Derivation*

Encoding elements for the NTAX information model EntityClass (`owl:Class`) and DisjointClasses (`owl:AllDisjointClasses`) components are listed in Table 9 and Table 10, respectively. The table format used to document the derivation of the content of these encoding elements is as follows:

- The **NTAX Modeling Concept** column specifies the class name, class attribute name, or class role name of the information modeling concept.
- The **NTAX Encoding Element** column specifies the OWL 2 or other Semantic-Web standard construct that is used to represent the corresponding NTAX Modeling Concept.
- The **NAS Modeling Concept** column specifies the NAS information modeling concept whose value is used (directly or indirectly) to derive the content of the corresponding concept in the NTAX.
- The **NAS Derivation Notes** column contains comments on the way in which the content value for an element is determined.

The NTAX Information Model elements for an EntityClass (`owl:Class`) are specified in Table 6 (see Section 5.4.3.3). The derivation of the content of these encoding elements is specified in Table 9.

[38] Some NAS classes are subclasses of classes defined in external standards. Reference to classes in external standards is excluded from the NTAX because those external standards are not available in OWL and therefore cannot be linked in the encoding.
[39] The visualization tool used for this diagram (OntoGraf plug-in in Protégé) uses downward-pointing arrows to represent the subclass relationship. This is different from the notation used in UML class diagrams.

Table 9 – Derivation of NTAX Encoding Elements for an Entity Class

NTAX		**NAS**	
Modeling Concept	**Encoding Element**	**Modeling Concept**	**Derivation Notes**
Entity Class	`owl:Class`	Entity Type Class	NTAX Entity Classes correspond to NAS Entity Types.
classIRI	`rdf:about`	[Indirect derivation using NSG Alphanumeric Code]	Content is the taxonomy-class IRI (see Sections 5.2.4.7 and 5.4.2.5). The NSG Alphanumeric Code is used as the terminal component of the IRI.
isAbstract	`iso19150-2:isAbstract`	Abstract?	Boolean content identical to NAS ‘Abstract?’.
label	`rdfs:label`	NSG Alphanumeric Code	Content identical to NAS Alphanumeric Code.
name	`skos:prefLabel`	Name	Content identical to NAS Name.
alias	`skos:altLabel`	Alias	Content identical to NAS Alias.
definitionNote	`skos:definition`	Definition Description	Incorporates the content of both the NAS Definition and NAS Description.
sourceIRI	`rdfs:isDefinedBy`	[Indirect derivation using Item Identifier]	Content is the URL value for the AS Registry entry defining the NAS concept. The URL incorporates the NAS Item Identifier as part of the query expression.
partOf	`dct:isPartOf`	[Not derived from NAS]	Content is the IRI value for the NSG Taxonomy. See Section 6.2.3.
Role name: generalization	`rdfs:subClassOf`	Entity Supertype Class	Indicates the Entity Type which is the generalization of the subject Entity Type.

The NTAX Information Model elements for DisjointClasses (`owl:AllDisjointClasses`) are specified in Table 7 (see Section 5.4.3.4). The derivation of the content of these encoding elements is specified in Table 10.

Table 10 – Derivation of NTAX Encoding Elements for Disjoint Classes

NTAX		**NAS**	
Modeling Concept	**Encoding Element**	**Modeling Concept**	**Derivation Notes**
DisjointClasses	`owl:AllDisjointClasses`	Unique Instance?	Boolean value of TRUE for ‘Unique Instance?’ of the Entity Supertype Class results in the creation of a DisjointClasses element containing the memberClass set of all the sibling Entity Type Classes to which the subject Entity Type Class belongs.
memberClass	`owl:members`	[Indirectly derived from the NAS Entity Type Class hierarchy]	Content is a set of two or more EntityClass IRIs (see classIRI) which collectively represent a set of NAS Entity Type Classes that all have the same direct Entity Supertype Class.

NOTE: The spelling conventions followed in NAS human-readable content are those of the Shorter Oxford English Dictionary, 6th Edition, Version 3.0.2.1. (NAS: Section 4.3)

Annex D – UML Primer

(Informative)

D.1 *UML Notations*

The diagrams that appear in this document are presented using the Unified Modeling Language (UML) static structure diagram with the ISO Interface Definition Language basic type definitions and the UML Object Constraint Language (OCL) as the conceptual schema language. The UML notations used in this Standard are described in Figure 15.

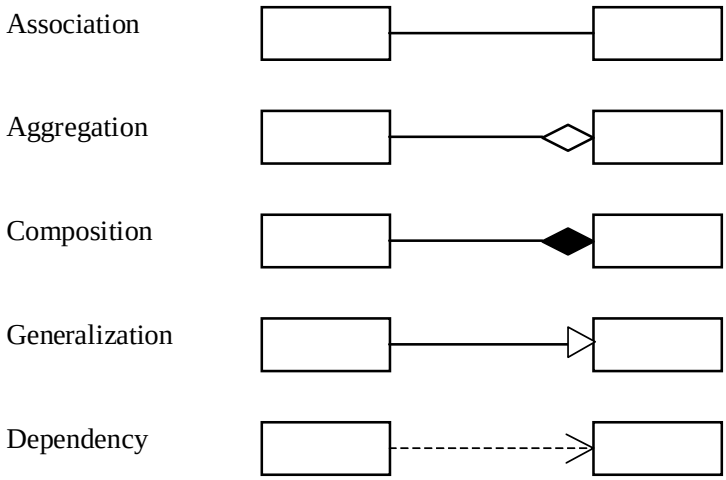

Figure 15 – UML Notation

D.2 *UML Model Relationships*

D.2.1 Associations

An association is used to describe a relationship between two or more classes. UML defines three different types of relationships, called association, aggregation and composition. The three types have different semantics. An ordinary association shall be used to represent a general relationship between two classes. The aggregation and composition associations shall be used to create part-whole relationships between two classes.

An aggregation association is a relationship between two classes in which one of the classes plays the role of container and the other plays the role of a containee.

A composition association is a strong aggregation. In a composition association, if a container object is deleted, then all of its containee objects are deleted as well. The composition association shall be used when the objects representing the parts of a container object cannot exist without the container object.

D.2.2 Navigation

Associations may be navigable in only one direction. If the direction is not specified, it is assumed to be a two-way association. If one-way associations are intended, the direction of the association can be marked by an arrow at the end of the line. Navigability means that instances participating in links at runtime (instances of an association) can be accessed efficiently from instances participating in links at the other end of the association. The precise mechanism by which such access is achieved is implementation specific. If an end is not navigable, access from the other ends may or may not be possible, and if it is, it might not be efficient.

D.2.3 Generalization

A generalization is a relationship between a superclass and the subclasses that may be substituted for it. The superclass is the generalized class, while the subclasses are specified classes.

D.2.4 Instantiation / Dependency

A dependency relationship shows that the client class depends on the supplier class/interface to provide certain services, such as:

- Client class accesses a value (constant or variable) defined in the supplier class/interface;
- Operations of the client class invoke operations of the supplier class/interface;
- Operations of the client class have signatures whose return class or arguments are instances of the supplier class/interface.

An instantiated relationship represents the act of substituting actual values for the parameters of a parameterized class or parameterized class utility to create a specialized version of the more general item.

D.2.5 Roles

If an association is navigable in a particular direction, the model shall supply a “role name” that is appropriate for the role of the target object in relation to the source object. Thus in a two-way association, two role names will be supplied. Figure 16 represents how role names and cardinalities are expressed in UML diagrams.

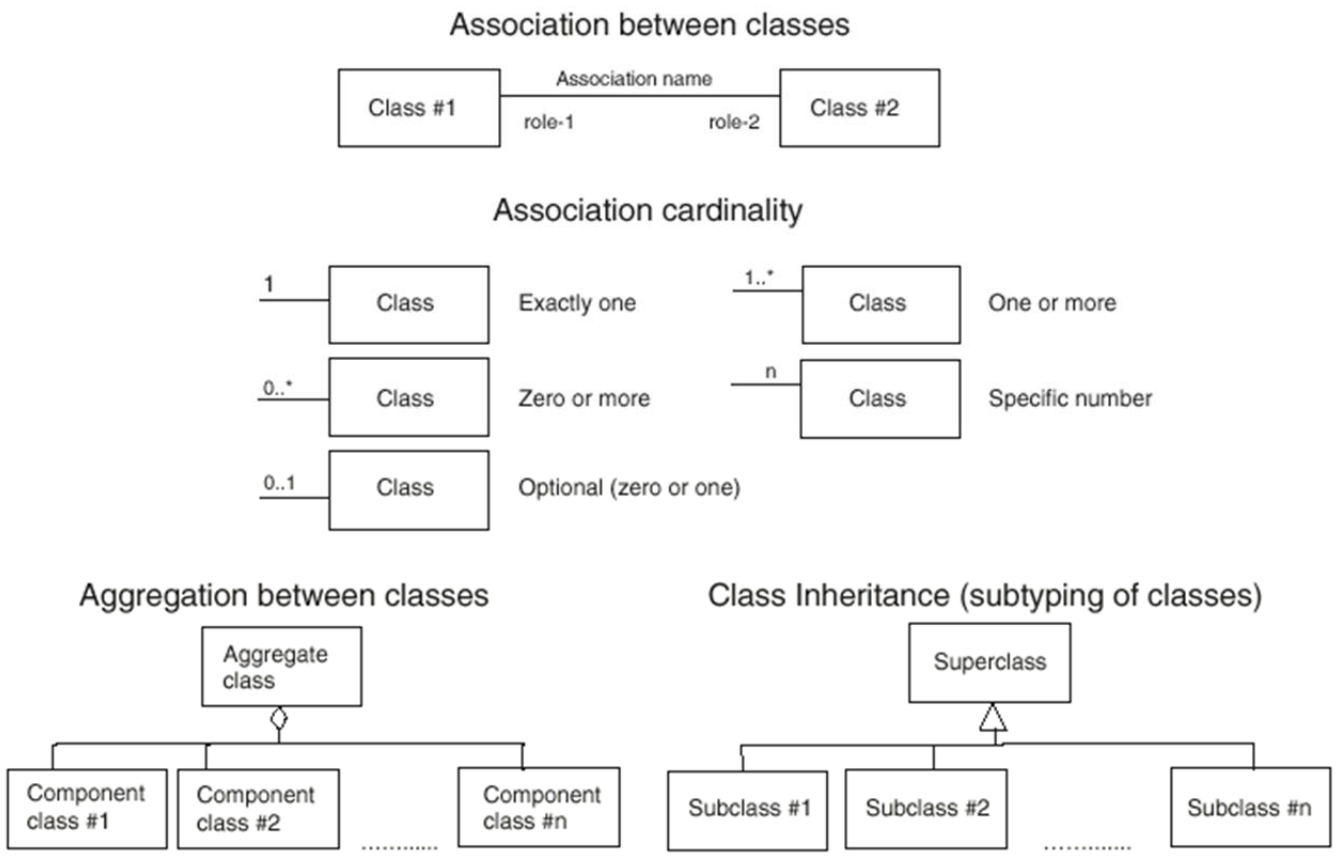

Figure 16 – UML Roles

D.3 *UML Model Stereotypes*

A UML stereotype is an extension mechanism for existing UML concepts. It is a model element that is used to classify (or mark) other UML elements so that they in some respect behave as if they were instances of new virtual or pseudo metamodel classes whose form is based on existing base metamodel classes. Stereotypes augment the

classification mechanisms on the basis of the built-in UML metamodel class hierarchy. Below are brief descriptions of the stereotypes used in this document.

In the NSG Application Schema, the following stereotypes are used:

a. <<type>> class used for specification of a domain of instances (objects), together with the operations applicable to the objects. A type may have attributes and associations.

b. <<enumeration>> datatype whose instances form a list of named literal values. Both the enumeration name and its literal values are declared. Enumeration means a short list of well-understood potential values within a class.

c. <<dataType>> a descriptor of a set of values that lack identity and whose operations do not have side effects. Datatypes include primitive pre-defined types and user-definable types. Pre-defined types include numbers, string, and time. User-definable types include enumerations.

d. <<codeList>> used to describe a more open enumeration. <<codeList>> is a flexible enumeration. Code lists are useful for expressing a long list of potential values. If the elements of the list are completely known, an enumeration should be used; if the only likely values of the elements are known, a code list should be used.

e. <<union>> describes a selection of one of the specified types. This is useful to specify a set of alternative classes/types that can be used, without the need to create a common super-type/class.

f. <<abstract>> class (or other classifier) that cannot be directly instantiated. The UML notation for this is to show the name in italics.

g. <<leaf>> package that contains definitions, without any sub-packages.