CDISC Standards in RDF Reference Guide

Version 1.0 Final

Prepared by

The PhUSE CS Semantic Technology Working Group

Notes to Readers

This document is the specification of the RDF vocabulary and the RDF data sets that represent the current foundational CDISC standards in a W3C endorsed semantic format.

Revision History

Date	Version	Summary of Changes
2014-12-12	Draft 1.0	Version 1.0 for public comment.
2015-06-18	Final 1.0	Version 1.0 Final

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This document provides a reference to the representation of the existing foundational CDISC standards in a model based on the Resource Description Framework (RDF). The RDF standard is a World Wide Web Consortium (W3C) Recommendation that today defines the backbone of the semantic web.

The document starts with a short introduction to the W3C semantic standards, followed by a detailed reference to the RDF schemas and models that were created to represent the CDISC standards CDASH, SDTM, SEND, ADaM, and their controlled terminologies in RDF. The key elements of the RDF meta-model are based on the ISO 11179 standard for Metadata Registries (MDR), a standard that also provides the background model for the CDISC metadata registry called SHARE.

A detailed walk-through for SDTM IG 3.1.3 illustrates the general structure of the RDF models.

1.1 Purpose

The CDISC mission states the following (emphasis added):

> The CDISC mission is to develop and support global, **platform-independent data standards** that enable **information system interoperability** to improve medical research and related areas of healthcare.

The CDISC Mission and Principles also state among others the following Core Principles:

- Recognize the ultimate goal of creating regulatory submissions that allow for flexibility in scientific content and are **easily interpreted**, **understood**, and **navigated** by regulatory reviewers.
- Acknowledge that the data content, structure and quality of the standard data models are of paramount importance, **independent of implementation strategy and platform**.
- Work with other professional groups to encourage that there is **maximum sharing of information** and minimum duplication of efforts.

The emphasized keywords illustrate the purpose of representing the existing CDISC standards in RDF. Designing data standards is hard, and designing platform-independent data standards that enable information system interoperability is even harder. CDISC has published a wide range of standards, but platform independence and interoperability, even between the standards, remain a formidable challenge. CDISC standards are designed in different formats, ranging from informal data models to XML specifications and UML models. CDISC standards are also published in different formats, such as XML schemas, PDF documents and Excel files.

The Resource Description Framework (RDF) provides a universal, mathematically precise, and computable language that can express a wide range of information. RDF can express information about meta-models, models, and data in the same universal language. Once expressed in RDF, information can be represented, accessed, computed, integrated, and exchanged without the need for any translations. This representation in RDF avoids information mismatches that often happen when systems interface with each other. RDF provides a consistent language and modeling framework, identical at design and run-time. As a W3C Recommendation, it can fulfill the role of a standard language to express and exchange standards.

RDF is also the foundation on which other W3C semantic standards are built (RDFS, OWL, and SKOS). Together they provide a rich medium to express vocabularies, taxonomies, ontologies, or knowledge representation in general. This supports the requirement for a data standard to enable interpretation, understanding, and navigation of information in a consistent manner. The SDTM IG 3.1.3 walk-through later in this document illustrates this point.

RDF has been designed from the start with platform independence and semantic interoperability in mind. As the backbone of the semantic web, it is targeted at distributing and integrating information across large scale heterogeneous networks. Publishing and sharing of information imply several virtues. The obvious one is to make the information available. Less obvious, but equally important, is to ensure that the information can be linked to and integrated with other information sources. Linked and integrated data truly maximizes the sharing of information and encourages reuse across information providers. RDF has been invented with exactly this purpose in mind.

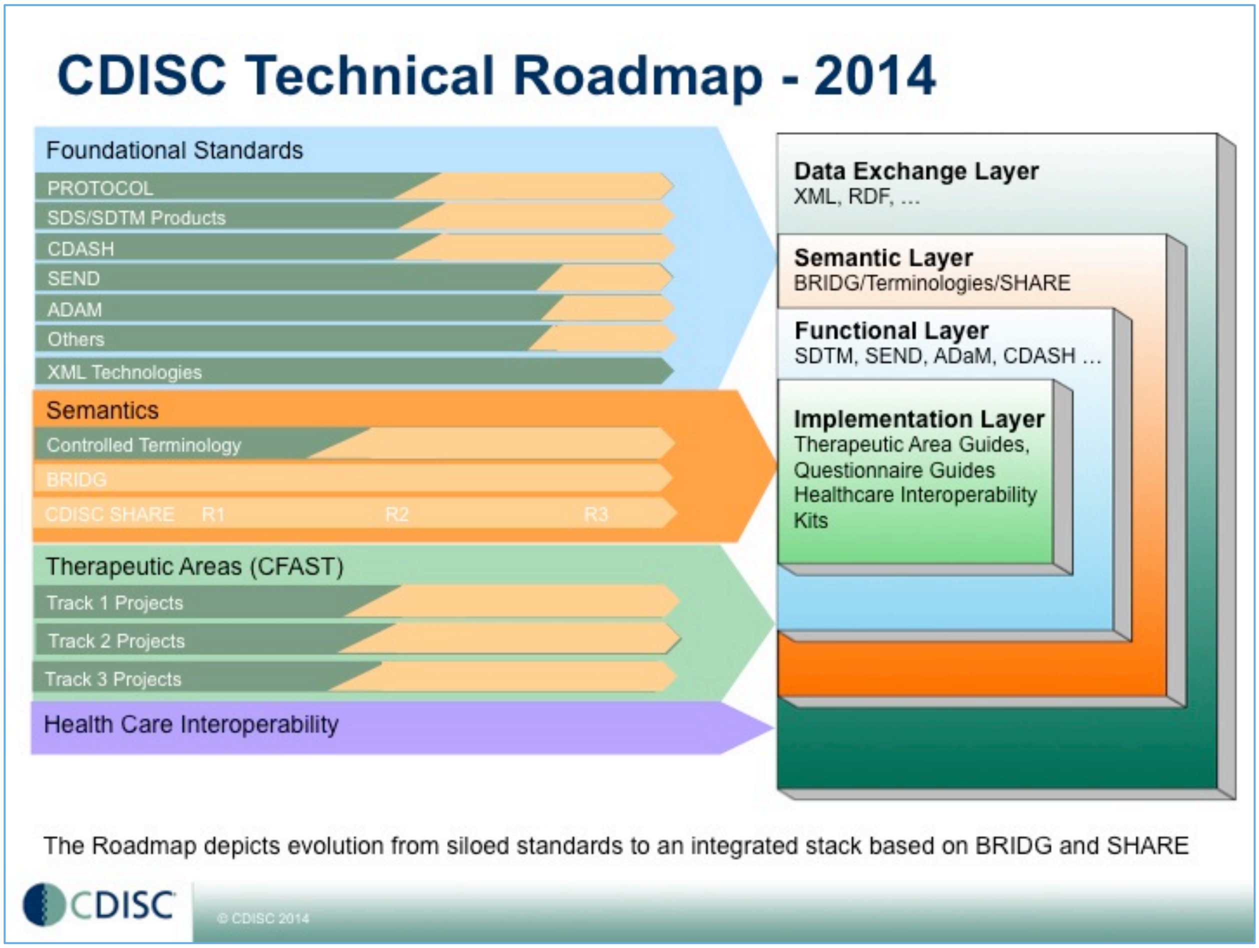
1.2 CDISC Roadmap

The 2014 CDISC Technical Roadmap (next page) includes RDF as part of a future exchange layer. The RDF models described in this guide represent the first step in that direction.

Future activities will be coordinated with planned work on SHARE in 2015. SHARE Release 3 includes requirements to deliver a comprehensive RDF export package from SHARE to support standards with semantic web tools and to enable end to end delivery of data standards.

Part of the work will focus on further alignment between the SHARE and RDF models, which are both based on the ISO 11179 Metadata Registry standard.

Work has also started by the PhUSE Semantic Technology working group to create a comprehensive RDF model that can represent protocol and study design. This work will be coordinated with the CDISC Protocol Representation Group and the TransCelerate Common Protocol Template working stream.



1.3 Acknowledgements

The project for creating RDF models for the existing CDISC standards involved collaboration and commitment from many volunteers. Contributions included providing expertise on RDF modeling, understanding the CDISC standards, discussing and resolving many modeling decisions, creating upload files for the RDF models, setting up the GitHub project, and writing this Reference Guide.

The following people, in alphabetical order, provided key contributions.

Phil Ashworth, Scott Bahlavooni, Daniel Boisvert, Susan DeHaven, Nathan Freimark, Josephine-Anne Gough, Laura Hollink, Dave Jordan, Ron Katriel, Kirsten Langendorf, Geoff Low, Frederik Malfait, Mitra Rocca, Gary Walker.

Also recognized are the efforts by Robert Wynne and Erin Muhlbradt from the NCI Enterprise Vocabulary Services (EVS) to publish the CDISC Controlled Terminology in RDF based on earlier discussions of the meta-model schema.

1.4 References

1. The starting chapter gives only a very brief introduction to semantic technologies. To read more on knowledge systems in general and semantic modeling in particular, the following references are recommended.
   - Dean Allemang and James Hendler. Semantic Web for the Working Ontologist. Second Edition. Morgan Kaufmann, 2011. This is an excellent book, well-written, specifically on the modeling aspects of RDF and OWL.
   - Christopher Walton. Agency and the Semantic Web. Oxford University Press, 2006. This book gives a broad outlook on knowledge systems and the semantic web, including more academic background on the computational aspects of the subject.
   - John Domingue, Dieter Fensel, and James Hendler. eds. Handbook of Semantic Web Technologies. Springer Reference. 2011. At over a 1000 pages, this Springer Reference work covers many aspects of the semantic web.
   - Dragan Gasevic, Dragan Djuric, and Vladan Devedzic. Model Driven Engineering and Ontology Management. Second Edition. Springer, 2009. This book provides valuable insight on knowledge engineering and the relationship between the different modeling spaces.
2. Here is a good entry page to locate the W3C standards for the semantic web, in particular RDF, RDFS, OWL, SKOS, and SPARQL:
   http://www.w3.org/2001/sw/wiki/
3. To see what the NCI Enterprise Vocabulary Services (EVS) is doing in the area of controlled terminologies and ontology modeling, have a look here:
   https://cabig.nci.nih.gov/concepts/EVS/
4. The National Center for Biomedical Ontology (NCBO) is a great resource for biomedical ontologies and related technologies. It provides a repository and federated search across a large number of biomedical ontologies. It can be accessed here:
   http://www.bioontology.org/
5. The CDISC standards reference documents for CDASH 1.1, SDTM 1.2, SDTM IG 3.1.2, SDTM 1.3, SDTM IG 3.1.3, SEND 3.0, ADaM 2.1, and ADaM IG 1.0 can be downloaded from the CDISC web site:
   http://www.cdisc.org/

1.5 Glossary

Term	Description
ADaM	Analysis Data Model, published by CDISC
AE	Adverse Events domain

Term	Description
BRIDG	Biomedical Research Integrated Domain Group, published by CDISC
CDASH	Clinical Data Acquisition Standards Harmonization, published by CDISC
CDISC	Clinical Data Interchange Standards Consortium
Concept	A unit of thought that can be abstracted as an element of a concept model
CRF	Case Report Form
CT	Controlled Terminology
Data Element	A data element represents a placeholder for a unit of data with a precise meaning defined by a corresponding data element concept
Data Value	A data value represents an element of the value range that can be assigned to a data element and has a precise meaning defined by a corresponding data value concept
DC	Dublin Core, a commonly used foundational RDF based vocabulary
DM	Demographics domain
EVS	Enterprise Vocabulary Services, provided by the National Cancer Institute
EX	Exposure domain
FDA	US Food and Drug Administration
First-order logic	A formal system to model individuals (objects) and predicates (relationships) between individuals
Git	Distributed revision control and source code management system
GitHub	Hosting service based on the Git revision control system at github.com
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
Inference	The action of deriving new triples from existing ones based on the axioms that govern the predicates from which the triples are built
IRI	Internationalized Resource Identifier
ISO 11179 MDR	ISO 11179 Metadata Registry standard
ISO 8601	ISO standard for date and time formats
Meta-model	A model intended to describe models
MOF	Meta Object Facility
NCBO	National Center for Biomedical Ontology
NCI	National Cancer Institute
Object	A literal or a URI referring to another RDF resource
ODM	CDISC Operational Data Model for exchanging clinical trial data and metadata
OMG	Object Management Group
Ontology	Used interchangeably for Semantic Model
OWL	Web Ontology Language, published by the W3C
OWL Profile	Subset of OWL that satisfies specific run-time requirements

Term	Description
PDF	Portable Document Format
Predicate	The aspect of the subject about which an RDF statement is made, usually a property or a relationship with another resource
RCS	Revision Control System
RDF	Resource Description Framework, published by the W3C
RDF/XML	XML serialization format of RDF
RDFS	RDF Schema, published by the W3C
SDTM	Study Data Tabulation Model, published by CDISC
SDTM 1.2	The SDTM 1.2 standard describing the high-level data structures for the CDISC data tabulation model
SDTM 1.3	The SDTM 1.3 standard describing the high-level data structures for the CDISC data tabulation model
SDTM IG 3.1.2	The SDTM Implementation Guide companion to SDTM 1.2 describing in detail the structure and guidance for submitting clinical trial data as a set of data domains
SDTM IG 3.1.3	The SDTM Implementation Guide companion to SDTM 1.3 describing in detail the structure and guidance for submitting clinical trial data as a set of data domains
Semantic Model	A model based on conceptual graphs or triples
SEND	CDISC standard for the exchange of non-clinical data
SHARE	The CDISC Metadata Registry (MDR) for managing and distributing the CDISC standards, also referred to as a Metadata Repository
SKOS	Simple Knowledge Organization Schema, published by the W3C
SPARQL	SPARQL Protocol and RDF Query Language, published by the W3C
Subject	RDF Resource about which a statement is made
TA	Trial Arms domain
Triple	A piece of knowledge in the form of a subject-predicate-object statement, fundamental building block of a semantic model
Triple Store	A database designed to hold sets of triples
Turtle	Serialization format of RDF
URI	Uniform Resource Identifier
URI Minting	Policy for creating URI elements
URL	Uniform Resource Locator, a URI that can be located on a network
XML	Extensible Markup Language, published by the W3C
XSD	XML Schema Definition
W3C	World Wide Web Consortium, standards organization publishing most of the well-known web standards, among others XHTML, XML, XML Schema, RDF, RDFS, OWL, SKOS, SPARQL etc.

1.6 Known Issues

This section summarizes in one place important known issues that are also mentioned elsewhere in the text.

SHARE and RDF

Work on the RDF representation of the foundational CDISC standards started before implementation of the first release of CDISC SHARE. Future SHARE releases will feature a large range of export formats, one of which will be RDF. Both approaches are based on the ISO 11179 model, but future alignment may result in deviations from the current package. Alignment and implementation of the RDF export will be part of SHARE Release 3, planned for 2015.

The current SHARE release contains a copy of BRIDG and relationships from foundational standards to BRIDG. Future releases of SHARE are expected to implement a comprehensive concept model and traceability between the different standards. These topics are not part of the current RDF package, but are good candidates for future inclusion. Note that the BRIDG 3.2 distribution contains an OWL model, but model that is not integrated in this RDF package. It is not certain if BRIDG will continue to support an OWL format in the future.

Not included in the current RDF package is the part of the ISO 11179 model that supports versioning and life cycle information of metadata items (called Administered Item in ISO 11179 and Asset in SHARE). Item level versioning is not part of the currently published CDISC standards. In the future, it will depend on the specific design and implementation in SHARE. When available in SHARE, it will be an important extension to the RDF export.

Normative Issues

The current RDF package represents the CDISC foundational standards as published and intends to respect the normative character of the standards with two exceptions.

The RDF model contains an additional attribute to express the data type of a data element using the W3C XML Schema Definition (XSD) standard. The XSD data types are more accurate than the published data types (e.g. SDTM Controlled Terms or Format), but cannot be considered to be normative. This is a topic for further consideration.

Assumptions in the implementation guides contain a wealth of information, but are scattered as text elements throughout the guides. The RDF model includes the assumptions that can be related to specific domains or data elements. The assumption text has been mapped verbatim to the RDF model, but the relationships between the text elements and specific domains or data elements have been mapped on a best effort basis and cannot be considered normative. This is a topic for further consideration.

2 W3C Semantic Standards Overview

The World Wide Web Consortium (W3C) has published a set of standards that define how to model, capture, and disseminate information with the explicit goal of maximizing semantic interoperability. This section gives a brief overview of these standards and explains how they provide a solid foundation to express the existing foundational CDISC standards in a more formal and precise way.

2.1 The Resource Description Framework (RDF)

RDF is the cornerstone of the W3C semantic standards. It is a general purpose framework for naming, describing, and organizing resources.

2.1.1 Resources

Within the existing web standards, the term resource refers to anything that can be identified by a Uniform Resource Locator (URL) and can be displayed in a web browser after issuing an HTTP request for that URL. This is the familiar way to request web pages, images, and other resources for viewing in a web browser. The representation of the resources that are delivered to the browser usually consists of HTML markup that contains a mix of textual information, rendering instructions, and hypertext links to other resources. This is how most of the web content is accessed and viewed today. Disseminating information in this way is very informal and ad-hoc. It requires human interpretation to identify information elements and the statements that are being made about them.

Within RDF, the term resource refers to something that somebody would like to talk about. A resource is identified and can be referenced by a Uniform Resource Identifier (URI). Unlike a URL, it is not required that a URI can be located and resolved over a network. Its purpose is mainly to name the things that are being talking about. Of course, if one decides to publish an RDF data set on the web, the most obvious way to make a representation of the resource available over HTTP is by making the URI resolvable, effectively turning the URI into a URL. This is why RDF resources usually have a URI with an HTTP scheme, even when they are not resolvable over a network. The use of a domain within such a scheme also indicates the resource owner and provides support for globally unique identifiers.

The following is an example of an RDF resource that represents the Adverse Event Outcome variable of the AE domain in the SDTM Implementation Guide 3.1.2:

```
http://rdf.cdisc.org/std/sdtmig-3-1-2#Column.AE.AEOUT
```

This URI has the following structure: the protocol scheme `http`, followed by the host name `rdf.cdisc.org`, the path `/std/sdtmig-3-1-2`, and the fragment identifier: `Column.AE.AEOUT`.

Just like in XML, the use of namespaces can provide a shorthand for this URI. After agreeing that the prefix `sdtmig-3-1-2` refers to the namespace

`http://rdf.cdisc.org/std/sdtmig-3-1-2#`

the same URI can be abbreviated as `sdtmig-3-1-2:Column.AE.AEOUT`. Such an abbreviation is known as a qualified name.

2.1.2 Triples

Within a web page, textual information about a resource is informal and left to the reader for interpretation. Within RDF there is a very precise way to make statements about resources. To give an example, let us start with the `sdtmig-3-1-2:Column.AE.AEOUT` resource introduced earlier. The SDTM Implementation Guide (IG) states that this variable has controlled terminology and that the code list for this variable is identified by the NCI EVS code C66768. In the IG this information is given as textual information within a PDF file, and the same information can be found in the SDTM spreadsheet files. The IG also states that the label of this variable is “Outcome of Adverse Event” and its name is “AEOUT”.

In RDF one can proceed as follows. Introduce a resource `sdtmct:C66768` to represent the code list C66768, a resource `mms:dataElementValueDomain` to represent the relationship between a data element and its code list, and the resources `mms:dataElementLabel` and `mms:dataElementName` to represent the label and name properties of a data element. For now just use these qualified names and don’t worry about the actual namespaces (`sdtmct` refers to a namespace for SDTM controlled terminology and `mms` refers to a namespace for the meta model schema of the CDISC foundational standards).

The IG information can now be represented in RDF as follows:

```
sdtmig-3-1-2:Column.AE.AEOUT mms:dataElementName "AEOUT" .
sdtmig-3-1-2:Column.AE.AEOUT mms:dataElementLabel "Outcome of Adverse Event" .
sdtmig-3-1-2:Column.AE.AEOUT mms:dataElementValueDomain sdtmct:C66768 .
```

Each line is an RDF statement and has three RDF terms. The first RDF term is called the *subject,* which refers to the resource about which a statement is made. The second RDF term is called the *predicate,* which identifies the aspect of the subject being talking about, like a property or a relationship with another resource. The third RDF term is called the *object* and can be either a literal or a URI referring to another resource. Since RDF statements take the format just described, they are also called triples.

Notes. (1) An RDF term may also be a blank node when only the existence of a resource needs to be expressed without naming it. (2) In RDF 1.1, published early 2014, a resource identifier is now an Internationalized Resource Identifier (IRI), which allows the use of Unicode characters. (3) In RDF 1.1, a literal *always* has a data type, which identifies the value space that the literal values can range over. RDF allows most of the XML Schema data types. If not specified, the data type of a literal is `xsd:string`. A string literal may optionally have a language tag, in which case the data type is `rdf:langString`.

In RDF, everything is expressed as a triple, and an RDF data set is a just a set of triples. You can think of RDF as a universal language that allows you to make any kind of statement about any kind of resource. The representation of the existing foundational CDISC standards in RDF moves the specification of these standards from a paper based format to a machine readable, computable, and semantic interoperable format.

2.1.3 Graphs

Representing subjects and objects of triples as nodes, and predicates of triples as directed edges between nodes, one can see that the information expressed by a set of triples is in fact a directed graph, which expresses the intrinsic data model of an RDF data set. The following example is part of the RDF representation of the SDTM IG 3.1.2.

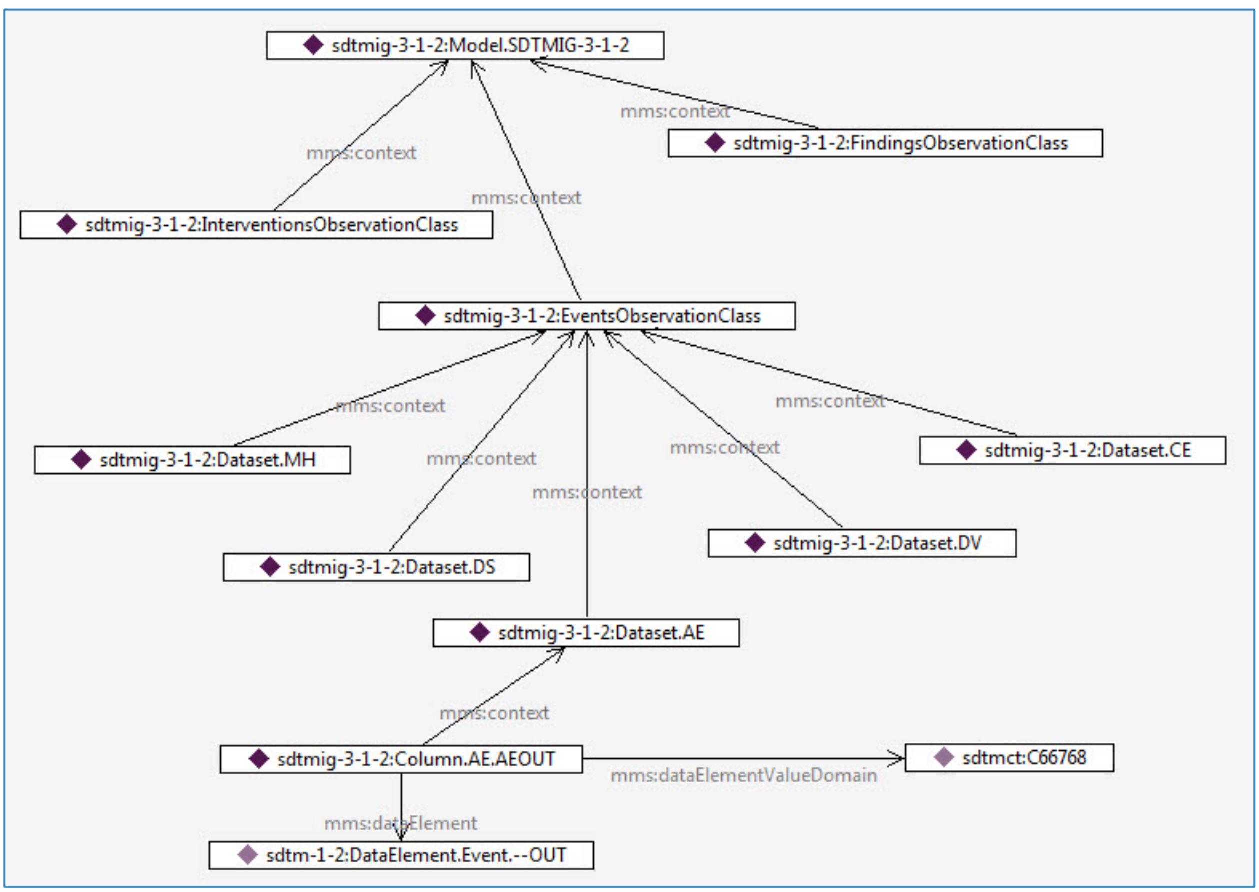


Figure 1. Example RDF Graph for SDTM IG 3.1.2

The graph data model is more versatile compared to tabular or hierarchical data models. It is also easy to integrate information from different RDF graphs, simply by taking the union of the graphs and by creating triples that link nodes across graphs. The use of connections within and across graphs leads to the notion of linked data. Figure 1. Example RDF Graph for SDTM IG 3.1.2 contains RDF terms from three different graphs. The `sdtmig-3-1-2` graph references the `sdtmct` graph by linking the AEOUT variable to its code list C66768. The `sdtmig-3-1-2` graph also references the `sdtm-1-2` graph by linking the AEOUT variable to its corresponding --OUT data element in SDTM 1.2.

The following is the list of graphs with their recommended namespace prefixes that have been created to represent the existing foundational CDISC standards in RDF.

Foundational CDISC Standards in RDF

```
cdash-1-1       http://rdf.cdisc.org/std/cdash-1-1#
sdtm-1-2        http://rdf.cdisc.org/std/sdtm-1-2#
sdtm-1-3        http://rdf.cdisc.org/std/sdtm-1-3#
sdtmig-3-1-2    http://rdf.cdisc.org/std/sdtmig-3-1-2#
sdtmig-3-1-3    http://rdf.cdisc.org/std/sdtmig-3-1-3#
adam-2-1        http://rdf.cdisc.org/std/adam-2-1#
adamig-1-0      http://rdf.cdisc.org/std/adamig-1-0#
sendig-3-0      http://rdf.cdisc.org/std/sendig-3-0#
```

CDISC Controlled Terminology in RDF published by the NCI Enterprise Vocabulary Services

```
cdashct     http://rdf.cdisc.org/cdash-terminology#
sdtmct      http://rdf.cdisc.org/sdtm-terminology#
adamct      http://rdf.cdisc.org/adam-terminology#
sendct      http://rdf.cdisc.org/send-terminology#
qsct        http://rdf.cdisc.org/qs-terminology#
glossary    http://rdf.cdisc.org/glossary-terminology#
```

2.2 Schemas and Ontologies

RDF is a standard language for naming, describing, and organizing resources in a graph-based data model using a set of RDF terms that constitutes a vocabulary. Information systems often have ways to express schemas and classifications about information elements, e.g. database systems have schemas that describe tables and columns, object oriented systems have classes and subclasses to describe common characteristics of objects, and programming languages have data types to express value sets of variables.

The W3C has defined a number of standard RDF vocabularies for organizing resources into types and expressing information about types and predicates. These vocabularies allow one to write models that express classifications and constraints about an RDF ‘fact base’ of resources. These RDF vocabularies are also called schemas or ontologies.

In particular, the W3C has defined the following standard schema languages:

- RDF Schema (RDFS)
- Web Ontology Language (OWL)
- Simple Knowledge Organization System (SKOS)

In the following sections, the features of these vocabularies that have been used writing the schemas for the representation of the foundational CDISC standards are introduced. For a full description to these vocabularies, the reader is referred to the references in Section 1.4.

2.2.1 RDF Schema (RDFS)

RDFS is a small vocabulary, written in RDF, to capture basic schema information about an RDF fact base.

```
rdf:type
```

The `rdf:type` predicate is part of RDF. It is used to assign types to resources and can be read as expressing the relationship "is a".

Example. The resource `sdtmct:C66768` in the SDTM controlled terminology is an enumerated value domain (code list). The resource `sdtmct:C66768.C48275` is a permissible value (code list element).

```
sdtmct:C66768 rdf:type mms:EnumeratedValueDomain .
sdtmct:C66768.C48275 rdf:type mms:PermissibleValue .
```

Resources that are assigned a type are called instances of that type. A type is also called a class, which acts as a set of instances.

```
rdfs:subClassOf
```

Types of resources are also called classes. The `rdfs:subClassOf` predicate expresses that one class is a subclass of another class, which is also called specialization. This predicate allows one to define class hierarchies to model specialization and generalization.

Example. The notion of an enumerated value domain specializes that of a value domain. One can also say that the notion of a value domain generalizes the notion of an enumerated value domain.

```
mms:EnumeratedValueDomain rdfs:subClassOf mms:ValueDomain .
```

```
rdfs:domain
```

The set of subjects for a given predicate is called its domain. The `rdfs:domain` predicate assigns a type to the domain of a predicate.

Example. If a predicate `mms:inValueDomain` relates a permissible value to an enumerated value domain then the domain of this predicate is the Permissible Value class.

```
sdtmct:C66768.C48275 mms:inValueDomain sdtmct:C66768 .
mms:inValueDomain rdfs:domain mms:PermissibleValue .
```

```
rdfs:range
```

The set of objects for a given predicate is called its range. The `rdfs:range` predicate assigns a type to the range of a predicate.

Example. If a predicate `mms:inValueDomain` relates a permissible value to an enumerated value domain, then the range of this predicate is the Enumerated Value Domain class.

```
sdtmct:C66768.C48275 mms:inValueDomain sdtmct:C66768 .
mms:inValueDomain rdfs:range mms:EnumeratedValueDomain .
```

If the range of a predicate is a set of literals, then one can use the XML Schema Definition (XSD) language to specify the literal data type.

Example. If a predicate `mms:dataElementName` relates a data element to its name, then the range of this predicate is the String data type. This data type is represented in RDF by the resource `xsd:string`.

```
sdtmig-3-1-2:Column.AE.AEOUT mms:dataElementName "AEOUT" .
mms:dataElementName rdfs:range xsd:string .
```

The namespace prefix `xsd` refers to `http://www.w3.org/2001/XMLSchema#`.

2.2.2 Web Ontology Language (OWL)

OWL is a more extensive W3C standard that provides a foundation to define more complex schemas called ontologies and to capture formal relationships that can be made subject to logical inference. In fact, the intrinsic meaning of each RDFS and OWL predicate, except for the annotation predicates, is completely determined in terms of logical inference.

Example. From the stated triples

```
sdtmct:C66768 rdf:type mms:EnumeratedValueDomain .
mms:EnumeratedValueDomain rdfs:subClassOf mms:ValueDomain .
```

an RDFS inference engine can derive the following triple

```
sdtmct:C66768 rdf:type mms:ValueDomain .
```

This is an application of the inference rule for `rdfs:subClassOf` saying that any element of a class is also an element of a parent class. The inference rule represents the intended meaning of the predicate.

The OWL standard defines an extensive set of predicates with a corresponding inference rule for each predicate, e.g. it is possible to model functional, inverse functional, symmetric, transitive, and many more types of predicates. The OWL standard also defines a number of profiles that define subsets of the full OWL standard, with each profile geared to satisfy specific requirements that are related to certain inference characteristics.

For the specification of the foundational CDISC standards in RDF, a very lightweight methodology was deliberately chosen. This decision eliminates the need for many of the OWL constructs, so that most of the modeling only uses OWL classes, sub-classes, domains, and ranges, thus avoiding the need to rely on inference logic.

2.2.3 Simple Knowledge Organization System (SKOS)

SKOS is another small RDF vocabulary that describes a basic language to organize concepts into concept systems, and to express relationships between concepts in terms of broader or narrower predicates. We have used mainly the annotation predicates of SKOS to annotate the schema elements within RDF. Annotations make the model completely self-documenting. The following annotation properties from RDFS and SKOS have been used throughout.

```
rdfs:label
```

This predicate provides a human-readable label of a resource.

Example. Administered Item is the root class for all the types of the meta model schema.

```
mms:AdministeredItem rdfs:subClassOf owl:Thing .
mms:AdministeredItem rdfs:label "Administered Item" .
```

It is considered good practice to provide an `rdfs:label` for every resource.


```
skos:definition
```

This predicate provides a statement of formal explanation of a resource.

Example. Classifiers are used as a generic mechanism to classify administered items.

```
mms:Classifier rdfs:subClassOf mms:AdministeredItem .
mms:Classifier skos:definition "A classifier defines a characteristic that
can be used to classify a set of administered items into a discrete family of
subsets." .
```

```
skos:example
```

This predicate provides a way to give further examples.

Example. Classifiers can be used to categorize SDTM variables such as the SDTM Core classification.

```
mms:Classifier rdfs:subClassOf mms:AdministeredItem .
mms:Classifier skos:example "SDTM classifies variables to be either required,
or expected, or permissible." .
```

```
skos:note
```

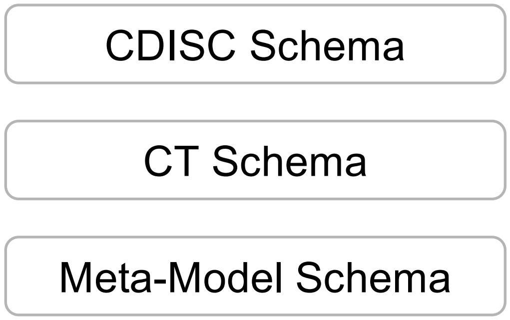
This predicate provides a way to specify additional notes.

Example. Classifiers are used as a generic mechanism to classify administered items.

```
cdiscs:DataElementCompliance rdfs:subClassOf mms:Classifier .
cdiscs:DataElementCompliance skos:definition "The compliance level of a CDISC
data element, e.g. required, permissible etc." .
cdiscs:DataElementCompliance skos:note "Some compliance levels have similar
labels but different meanings across the different CDISC standards. In those
cases different resources should be created to represent these." .
```

3 RDF Schemas for CDISC Foundational Standards

Three RDF schemas have been defined and layered that enable the description of the CDISC foundational standards for CDASH, SDTM, SEND, and ADaM, together with their controlled terminologies.



- The meta-model schema (namespace prefix `mms`) is a generic RDF schema with elements from ISO 11179 that allows the specification of data oriented models in a unified way.
- The CT schema (namespace prefix `cts`) defines additional predicates used by the NCI EVS to publish the CDISC controlled terminology in RDF.
- The CDISC schema (namespace prefix `cdiscs`) introduces additional classes and predicates to capture CDISC specific model information.

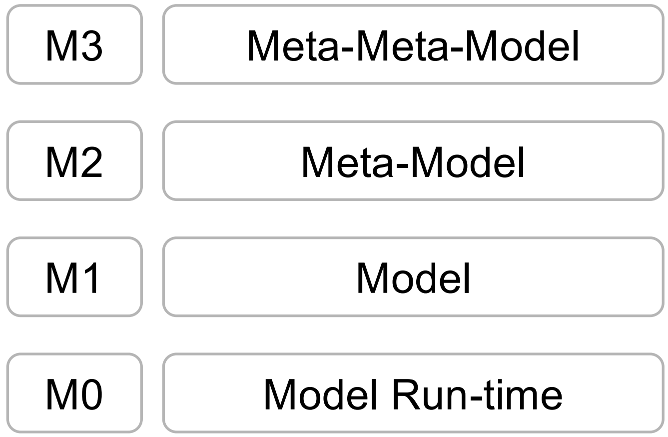
These schemas have been layered, i.e. the CDISC Schema imports the CT Schema, which in its turn imports the Meta-Model Schema. It is therefore sufficient to use the CDISC Schema to have all the schema elements available when defining a new CDISC model. The following sections give detailed descriptions for each of these schemas.

3.1 The Meta-Model Schema

A meta-model schema is a schema designed to specify models. We give a short introduction to the Object Management Group (OMG) framework and a more detailed introduction to the ISO 11179 standard for metadata registries (MDR). This section concludes with a complete description of the Meta-Model Schema.

3.1.1 OMG Meta Object Facility (MOF)

When talking about data standard specifications such as the CDISC standards for collecting, tabulating, and analyzing clinical and non-clinical trial data, it is important to understand that a description of these standards is in fact a meta-data description, i.e. information about the specification of a data standard. The Meta Object Facility (MOF), the OMG framework that describes the meta-model hierarchy used in several of the OMG standards, allows any number of layers, but the following 4-layer meta-model hierarchy serves to clarify the levels of discourse encompassed by the CDISC Meta Model schema.



- The model run-time (M0) contains the run-time instances of a model. In our case M0 corresponds to clinical trial data, either in its collected, tabulated, or analyzed format.
- The model layer (M1) defines constraints on the run-time instances. In our case M1 corresponds to the actual data models defined by the CDASH, SDTM, SEND, and ADaM data standards.
- The meta-model layer (M2) defines a language for specifying models. In our case M2 corresponds to the Meta-Model-Schema that is based on the ISO 11179 standard for metadata registries (MDR). This schema defines a unified meta-model to describe the CDISC data standards.
- The Meta-Meta-Model (M3) defines a language for specifying meta-models. In our case M3 corresponds to using RDF as the language for specifying the meta-model schema.

3.1.2 ISO 11179 for Metadata Registries

The ISO 11179 standard for Metadata Registries (MDR) is a very comprehensive M2-Level standard that describes the elements of a metadata registry, an asset intended to contain information about data elements and its associated metadata. ISO 11179 is published in six parts.

- Part 1: Framework
- Part 2: Classification
- Part 3: Registry meta-model and basic attributes
- Part 4: Formulation of data definitions
- Part 5: Naming and identification principles
- Part 6: Registration

ISO 11179 defines a common language to talk about different aspects of metadata registries and enables the exchange of such metadata between systems that follow this standard. The background model for CDISC SHARE is also based on ISO 11179, so that information exchange can be expected to be smooth. Expressing the ISO 11179 language elements in RDF has the benefit to make this information computable and semantically interoperable.

The ISO 11179 standard is a very extensive standard and it has not been the intention to provide a full representation of the standard in the RDF meta-model schema. Rather, focus has been on those elements of Part 3 that are sufficient to represent the CDISC standards in RDF in a unified way.

For the implementation in actual systems, metadata curators need to pay close attention to governance, which includes workflow processes to register and maintain metadata elements, provenance about the quality of the metadata elements, and life cycle information about the versioning of the metadata elements. Registration information is described in ISO 11179 Part 6, but has not been included in the current meta-model schema since this information is not available from the currently published CDISC standards. Once CDISC standards are published from SHARE, there is an opportunity to extend the RDF meta-model schema to also include life cycle information.

3.1.3 ISO 11179 Part 3 Registry Meta-model and Basic Attributes

The class diagram in Figure 2. ISO 11179 Part 3 Class Diagram integrates the core classes from ISO 11179 Part 3. The diagram is divided in two parts. The lower part represents the part of the model that deals with operational metadata. The upper part represents the conceptual layer. Operational metadata elements can be mapped to corresponding concepts in the concept model where they can be further modeled as or mapped into existing concept systems.

Operational Metadata

The classes below the line in the diagram are used to define operational models such as CDASH, SDTM, and ADaM. Typically, there are **Data Element** instances such as the SDTM variable AEOUT, which have a **Value Domain** that describes the value space of the data element. For discrete value spaces, an **Enumerated Value Domain** can be defined that consists of **Permissible Value** instances such as the value “Resolved”. Data elements can be collected in a **Context**, and contexts themselves can be grouped recursively in higher level contexts, e.g. the AEOUT variable is part of the AE domain, which in turn is part of the Event Observation Class. See also Figure 1. Example RDF Graph for SDTM IG 3.1.2 for an example with additional context relationships.

A data element that is collected in a context may have context dependent properties that are represented in a **Data Element Context**.

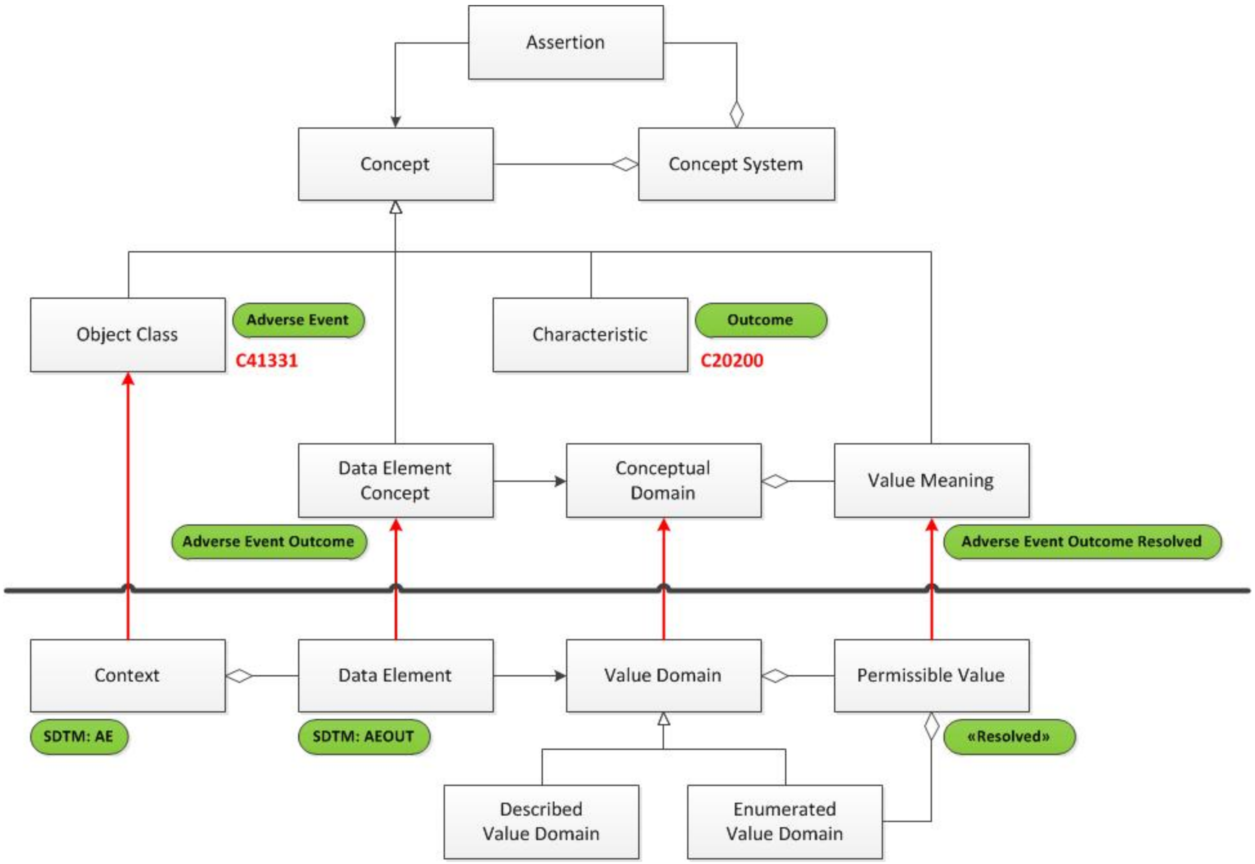


Figure 2. ISO 11179 Part 3 Class Diagram

This part of the model defines a schema to capture the operational elements of the metadata model. The classes of this part of the model have been represented as OWL classes in the RDF meta-model schema. They are sufficient to represent the CDISC operational standards as sets of data elements, organized in a hierarchy of contexts, with data elements linked to value domains where applicable.

Conceptual Metadata

The classes above the line in the diagram represent the conceptual layer of ISO 11179 Part 3. Typically, a data element is mapped to a **Data Element Concept**, e.g. AEOUT is mapped to the concept Adverse Event Outcome, and a permissible value is mapped to a **Value Meaning**, e.g. the value "Resolved" is mapped to the corresponding concept Adverse Event Outcome Resolved. Usually, a data element concept can be decomposed into an **Object Class** and a **Characteristic**, e.g. the concept Adverse Event Outcome can be decomposed into the concepts Adverse Event and Outcome. Earlier versions of ISO 11179 talked about Property instead of Characteristic. Another common synonym is attribute.

This set of classes is the minimal setup to define a metadata registry model with a basic concept layer that captures the core semantics of the operational models by mapping operational elements to conceptual elements.

This part of the model has not been mapped to the RDF meta-model schema because the CDASH, SDTM, SEND, and ADaM are operational data standards. Once conceptual metadata becomes available from CDISC SHARE in a systematic way, there is an opportunity to extend the RDF meta-model schema to include an ISO 11179 compatible concept layer.

Concept Systems

An additional feature that can be introduced is the mapping of conceptual elements to entries in external concept systems such as taxonomies, thesauri, ontologies, and other knowledge systems, e.g. the concept Adverse Event can be identified with the C41331 entry in the NCI Thesaurus, the concept Outcome can be identified with the entry C20200. The NCI Thesaurus is today already published in RDF. When ISO 11179 concepts are also represented in RDF, then linking both sources is only a matter of creating additional triples.

Data Element Concept, Value Meaning, Object Class and Characteristic are all subclasses of **Concept**. ISO 11179 defines additional classes to do more extensive concept modeling with formal semantics. The key notion is an **Assertion** that encapsulates a statement about one or more concepts in a **Concept System**. ISO 11179 has a few more classes to get this done (e.g. Relationship, Link) and provides examples in Part 3 Annex E mapping SKOS and OWL into ISO 11179. It shows how assertions can be used to capture RDF triples, where concepts can take the role in either of these components. Since the meta-model schema is already based on RDF, there is no need to create additional schema elements to cover this part of ISO 11179.

3.1.4 RDF Meta-Model Schema: Classes and Predicates

The OWL class hierarchy of the meta-model schema is shown in Figure 3. Meta-model Class Hierarchy.

Administered Item is the root class. In ISO 11179 an administered item is any item that has registration, governance, and life cycle information associated with it. The meta-model schema also has a number of custom specializations of **Context** such as the **Dataset** context to distinguish the different context levels in a model. Instances of **Model** represent top-level contexts.

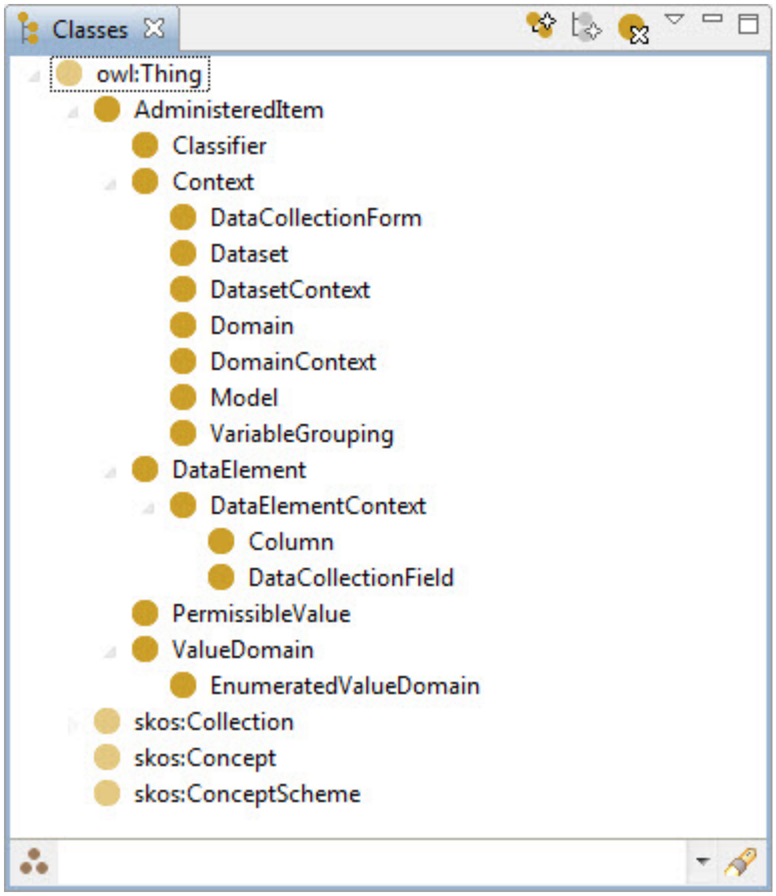


Figure 3. Meta-model Class Hierarchy

The following table provides details about the meta-model classes and predicates.

OWL Class

Administered Item	Any resource for which administrative information is recorded in an administration record.
Classifier	A classifier defines a characteristic that can be used to organize a set of administered items into a discrete family of subsets, e.g. SDTM classifies variables to be either required, or expected, or permissible.
Context	Each administered item is named and defined within one or more contexts. A context defines the scope within which the subject data has meaning. A context may be a business domain, an information subject area, an information system, a database, file, data model, standard document, or any other environment determined by the owner of the registry. Each context is itself managed as an administered item within the registry and is given a name and a definition.
Data Collection Form	Context to group a set of data collection fields.
Dataset	Context to group a set of dataset variables, e.g. SDTM IG Adverse Event domain.

OWL Class

Dataset Context	Context to group related data sets, e.g. Events Observation Class.
Domain	Context to group a set of related data elements.
Domain Context	Context to group related domains.
Model	Top-level context for all model elements, e.g. SDTM 3.1.2.
Variable Grouping	Context to group a set of data elements, e.g. SDTM Timing Variables.
Data Element	Unit of data that is considered in context to be indivisible.
Data Element Context	A data element context maps a data element to a context and describes context dependent properties of that data element.
Column	A column is a data element in the context of a dataset and may add context dependent properties such as the ordinal number within the dataset.
Data Collection Field	A data collection field is a data element in the context of a data collection form and may add context dependent properties such as optional or required.
Permissible Value	A permissible value is a designation, binding of signs (value) to a corresponding value meaning.
Value Domain	A value domain is a collection of permissible values. A value domain provides representation, but has no implication as to what data element concept the values are associated with, nor what the values mean.
Enumerated Value Domain	A value domain that is specified by a list of all its Permissible Values.

Properties for Administered Item

Context	The context of an administered item.
Identifier	The system identifier of an administered item
Ordinal	The ordinal of an administered item within a list of items.

Properties for Context

Context Description	The descriptive text documenting a context.
Context Label	Short descriptive label of a context.
Context Name	The name of a context that uniquely identifies it within the registry.

Properties for Data Element

Broader	Relates a data element to another data element that is more general, e.g. the data element AE.AEOUT in the SDTMIG 3.1.2 “has broader” the data element --AEOUT in SDTM 1.2.
Data Element Description	The descriptive text documenting a data element.

Properties for Data Element

Data Element Label	Short descriptive label of a data element.
Data Element Name	The name of a data element that uniquely identifies it within a context. A distinction must be made between the unique URI identifying a data element as a resource within the ontology and the name of a data element used to identify it within a context of a metadata registry.
Data Element Type	The type of a data element, i.e. the values it can range over. The actual value space may be further restricted by the specification of a value domain.
Data Element Value Domain	The value domain of a data element.

Properties for Data Element Context

Data Element	The data element that a data element context refers to.

Properties for Permissible Value

In Value Domain	A permissible value is a designation, binding of signs (value) to a corresponding value meaning.

Properties for Enumerated Value Domain

Subset Of	Identifies the parent enumerated value domain of which this enumerated value domain is a subset of, e.g. CDASH code lists are defined as subsets of SDTM code lists.

Table 1. Meta Model Schema

3.2 The CT Schema

The CDISC Controlled Terminology is published by the NCI Enterprise Vocabulary Services (EVS) in a number of formats including RDF. The CT schema imports the meta-model schema, adds no further classes, but adds predicates that reflect the properties used in the other formats. These schemas are consistent with the ones that are already being used by the NCI EVS.

Properties for Permissible Value and Enumerated Value Domain

CDISC Definition	Identifies the CDISC definition for a particular term. In many cases an existing NCI definition has been used. The source for a definition is noted in parentheses (e.g. NCI, CDISC glossary, FDA).
CDISC Submission Value	As per SDTM Implementation Guide this is the specific value expected for submissions.
CDISC Synonyms	Identifies the applicable synonyms for a CDISC Preferred Term. This is especially

Properties for Permissible Value and Enumerated Value Domain	
	important in instances where a Test name or Parameter Test name contains a corresponding Test Code or Parameter Test Code. Examples include: TSPARM, VSTEST, LBTEST and EGTEST.
NCI Code	Unique numeric code randomly generated by NCI Thesaurus (NCIt) and assigned to individual CDISC controlled terms.
NCI Preferred Term	This is the CDISC preferred name for a term as identified in the NCI Thesaurus (NCIt) environment and to which numeric codes are assigned. In most cases, the Preferred Term or PT is redundant with the CDISC Submission Value.

Properties for Enumerated Value Domain	
Codelist Name	Contains the descriptive name of the codelist which is also referred to as the codelist label in the SDTM Implementation Guide.
Is Extensible Codelist	Defines if controlled terms may be added to the codelist. New terms may be added to existing codelist values as long as they are not duplicates or synonyms of existing terms. The expectation is that sponsors will use the published controlled terminology as a standard baseline and codelists defined as extensible may have terms added by the sponsor internally.

Table 2. Controlled Terminology Schema

3.3 The CDISC Schema

The CDISC Schema imports the CT Schema, which in turn imports the meta-model schema. It is therefore sufficient for a new data model specification to import the CDISC Schema to have all the schema elements available.

The CDISC Schema adds a number of a classes that are specific to the CDISC standards as a whole or specific to a particular standard, thus avoiding the need to create additional schemas for each of the CDISC operational standards. Figure 4. CDISC Schema Class Hierarchy shows the class hierarchy of the meta-model and CDISC schema.

The most notable additions are classes to capture the assumptions of the implementation guides and a number of data element classifiers.

The assumptions are usually found in free text format following the description of a CDISC domain in the CDISC standards documents, but don’t always consistently describe these assumptions. Assumptions have been mapped to RDF on a best effort basis and where possible linked to the data elements or to a more general context to which they apply.

The character and numeric data types are used across multiple CDISC standards and have been defined as instances of `cdiscs:DataElementType`. Other classifiers only apply to specific CDISC standards and have been defined in their respective RDF models.

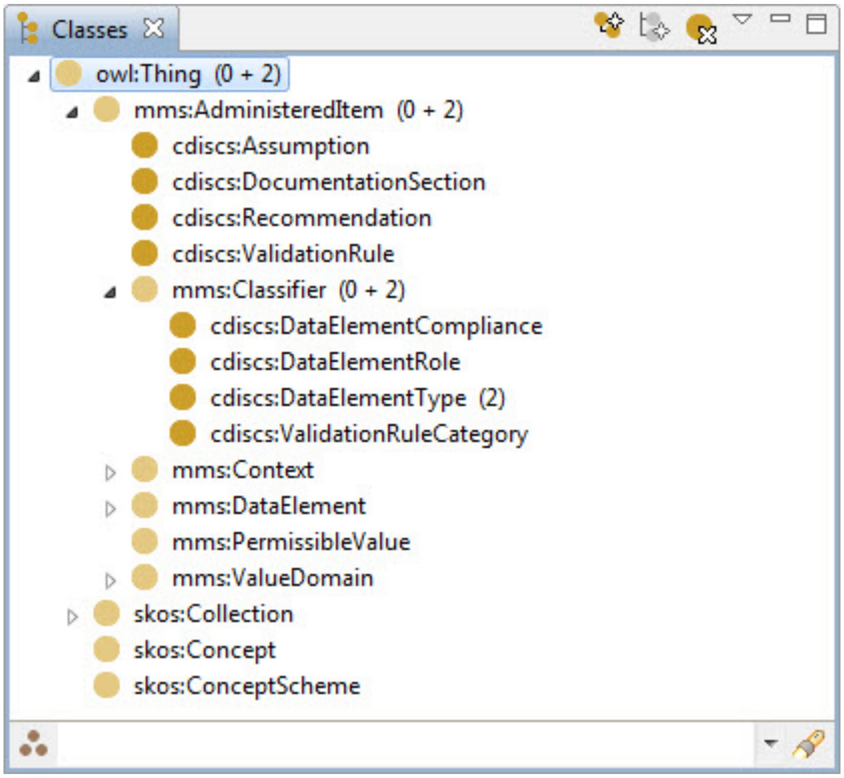


Figure 4. CDISC Schema Class Hierarchy

The CDISC Schema adds the following OWL classes, predicates, and instances.

OWL Class

Assumption	An assumption about an administered item specified by the CDISC documentation.
Documentation Section	A section that contains a set of assumptions about an administered item specified by the CDISC documentation.
Recommendation	A recommendation about an administered item specified by the CDISC documentation.
Validation Rule	A rule that can be used to validate the content of a CDISC data set against specifications given in the CDISC standard.
Data Element Compliance	The compliance level of a CDISC data element, e.g. required, permissible etc. Some compliance levels have similar labels but different meanings across the different CDISC standards. In those cases different resources should be created to represent these.
Data Element Role	The role of a CDISC data element, e.g. SDTM timing variable etc.
Data Element Type	The CDISC designated data type of a data element, character or numeric.
Validation Rule Category	The classification of a validation rule according to a given set of validation rule categories.

Instances of Data Element Type

Classifier Character	Classifies a CDISC data element of type character.
Classifier Numeric	Classifies a CDISC data element of type numeric.

Properties for Assumption

About	The administered item that this assumption is about.
Assumption Text	The text of an assumption.
Part of Section	The documentation section that this assumption is a part of.

Properties for Documentation Section

Documents	The administered item that this documentation section gives more details about.
Section Label	The label of a documentation section.

Properties for Recommendation

Recommendation Text	The text of a recommendation.

Properties for Dataset

Dataset Code	Each domain dataset is distinguished by a unique, two-character code that should be used consistently throughout the submission.
Dataset Structure	The record structure of a CDISC dataset, e.g. the structure of the SDTM AE dataset is “one record per adverse event per subject”.

Properties for Data Element

Completion Instructions	The completion instructions for a data element captured on a data collection form.
Controlled Terms or Format	A CDISC defined description about the controlled terms or the format of a data element in the implementation guide.
Data Element Compliance	Refers to the compliance level of a CDISC data element.
Data Element Role	Refers to the role of a CDISC data element.
Data Element Type	Refers to the type of a CDISC data element, character or numeric.
Prompt	The prompt for a data element captured on a data collection form.
Qualifies	Relates a qualifying data element to a qualified data element, e.g. the Findings data element --DECOD qualifies the data element --TERM.
Question Text	The question text of a data element captured on a data collection form.

Properties for Data Element	
References	References to sections of the printed CDISC documentation.
Supported by SDTMIG	Indicates if a data element is supported by the SDTM Implementation Guide (IG).
Supported by SEND	Indicates if a data element is supported by the Standard for Exchange of Non-clinical Data (SEND).

Table 3. CDISC Schema

4 RDF Datasets for CDISC Foundational Standards

The RDF schemas for CDISC Foundational Standards define an ISO 11179 based vocabulary and a common meta-model to express the content of the CDISC Foundational Standards for data collection (CDASH), data tabulation (SDTM, SEND), and data analysis (ADaM). This section describes how each of these CDISC standards is mapped to an RDF dataset.

4.1 CDASH 1.1

The CDASH 1.1 RDF dataset imports the CDISC schema, the CDASH Controlled Terminology dataset, and the SDTM Controlled Terminology RDF dataset. A CDASH code list is a subset of a corresponding SDTM code list.

The following table describes the classes used by CDASH 1.1.

OWL Class	Usage and Instantiation
Model	The CDASH Model is an instance of the `mms:Model` class which is a sub-class of `mms:Context`. Properties have been defined to represent the name, label, and description of the model. Variable Groupings such as Identifier, Timing, and Intervention Variables are incoming references to the CDASH Model.
Data Element Compliance	The Data Element Compliance class represents the Core Designations for Basic Data Collection Fields as presented in the CDASH specification. The model has properties for the compliance label and the compliance definition.
Domain Context	The Domain Context class represents the relationship between the CDASH Domains and the SDTM Observation Class. This relationship models the implicit assumption that an SDTM Data Element can be used at the implementer's discretion when producing a CDASH compliant Data Collection Form. There are two special purpose Domain Contexts that cover the Common Data Element and Common Timing Data Element, both of which may be used in any CDASH Domain. The model defines properties for the name, label, and description of the Domain Context.
Domain	The Domain class represents the Data Element Domain context. The model defines properties for the ordinal, the name (using the two character domain designation whenever applicable), and the label of the Domain.
Data Element	The Data Element class represents the definition of a CDASH Data Element. The model defines the following properties for a Data Element: ▪ Name: SDTM or CDASH Variable Name ▪ Label: CDASH Prompt ▪ Description: definition of the Data Element from the CDASH Specification ▪ Type: type definition of the Data Element ▪ Value Domain: reference to the associated Value Domain ▪ Controlled terms or Format: reference to the CDISC defined moniker ▪ Question Text: CRF text ▪ Completion Instructions: CRF completion instructions

OWL Class	Usage and Instantiation
	The model defines the Data Element Types using the more prescriptive XML Schema Definition (XSD) data types including `xsd:string`, `xsd:decimal`, `xsd:date`. These data types provide more accurate type information compared to the CHAR/NUM designations used in SDTM.
Data Collection Form	The Data Collection Form class represents the Domain sub-grouping referred to as Scenarios in the CDASH Specification. The model defines properties for the form name and the form label.
Data Collection Field	A Data Collection Field represents a single Data Element for representation in a Data Collection Form. It links a Data Element to a Data Collection Form and can be visualized in the CDASH specification as a Row in one of the Domain Tables. The model defines the following properties for a Data Collection Field: ▪ Ordinal: the expected order on a Data Collection Form ▪ Form: reference to the form containing the field ▪ Data Element: reference to the data element being collected ▪ Data Element Compliance: the Core Designation defined by CDASH
Documentation Section	The Documentation Section represents a Section within the CDASH specification document. The model defines the following properties for a Documentation Section: ▪ Documents: reference to the resource being documented ▪ Label: the section title
Recommendation	A Recommendation resource holds any guidance provided by CDISC as part of the CDASH Specification for users. Each Recommendation has a link to its parent Documentation Section. The Recommendation class is where the Information For Sponsors content from the CDASH specification is held. The model defines the following properties for a Recommendation: ▪ Ordinal: orders each Recommendation within a Section ▪ Recommendation related to: a reference to the relevant Administered Item within the specification such as a Data Element or a Domain ▪ Text: the content of the Recommendation There was a conscious effort to ensure that recommendations were streamlined. Some examples from the CDASH specification were not included as they did not add any further clarity to the intent of a section.

Table 4. OWL Classes for CDASH

4.2 SDTM 1.2 and SDTM 1.3

The SDTM 1.x RDF datasets imports the CDISC schema. The following table describes the classes used by SDTM 1.x.

OWL Class	Usage and Instantiation
Model	The SDTM Model is an instance of the `mms:Model` class which is a sub-class of `mms:Context`. Properties have been defined to represent the name, label, and description of the model. Variable Groupings such as Identifier, Timing, and Intervention Variables are incoming references to the SDTM Model.
Variable Grouping	Variable Groupings defined in SDTM are instances of `mms:VariableGrouping` which is a sub-class of `mms:Context`.Each Variable Grouping has a label and an ordinal property. SDTM variables such as STUDYID and USUBJID are incoming references to the Variable Groupings.
Variable	Variables defined within a Variable Grouping are instances of the class `mms:DataElement` and have the following properties: ▪ Variable Name ▪ Variable Label ▪ Variable Type as defined in SDTM ▪ Variable Type as defined by an XSD data type ▪ Variable Role ▪ Variable Description ▪ Order of the Variable within the Variable Group ▪ Variable Supported by SDTM IG Indicator ▪ Variable Supported by SEND IG Indicator Variables have no incoming references
Variable Role	A Variable Role is an instance of the `cdiscs:DataElementRole` class which is a sub-class of `mms:Classifier`. The following properties are defined: ▪ Variable Role Label ▪ Variable Role Definition Variables of a given Variable Role are incoming references to Variable Roles, e.g. --CAT and --SCAT variables are incoming references to the Grouping Qualifier role.
CDISC Core Compliance	Each CDISC Core Compliance (e.g. required, expected, permissible) is an instance of the class `cdiscs:DataElementCompliance` which is a sub-class of `mms:Classifier`. There are properties to describe the compliance label and the compliance definition.

Table 5. OWL Classes for SDTM

4.3 SDTM IG 3.1.2 and SDTM IG 3.1.3

The SDTM IG 3.x RDF dataset imports the SDTM 1.x and SDTM Controlled Terminology.

OWL Class	Usage and Instantiation
Model	The SDTM IG Model is an instance of the `mms:Model` class which is a sub-class of `mms:Context`. Properties have been defined to represent the name, label, and description of the model. Observation Classes are incoming references to the SDTM IG Model.

OWL Class	Usage and Instantiation
Dataset Context	Observation Classes are instances of `mms:DatasetContext` which is a sub-class of `mms:Context`. The following properties are defined: ▪ Observation Class Name ▪ Observation Class Label ▪ Observation Class Description ▪ Order of the Observation Class within the SDTM IG SDTM domains within an Observation Class are incoming references to the Observation Class, e.g. CM links to the Interventions Observation Class.
Dataset	SDTM domains are instances of the `mms:Dataset` class which is a sub-class of `mms:Context`. The following properties are defined: ▪ Domain Name ▪ Two-letter Domain Code ▪ Domain Label ▪ Domain Structure ▪ Order of the Domain within the Observation Class Variables within an SDTM domain and domain level assumptions are incoming references to an SDTM domain.
Column	SDTM variables are instances of the class `mms:Column` and have the following properties: ▪ Variable Name ▪ Variable Label ▪ Variable Type as defined in SDTM ▪ Variable Type as defined by an XSD data type ▪ IG Controlled Terms or Format ▪ Codelist URI defined by the NCI EVS ▪ Variable Role ▪ CDISC Notes ▪ CDISC Core Compliance ▪ CDISC Documentation References ▪ Order of the Variable within the Domain ▪ Data Element reference in SDTM 1.x Variable level assumptions specified below the IG domain specification are incoming references to the SDTM variable. The model defines the Data Element Types using the more prescriptive XML Schema Definition (XSD) data types including `xsd:string`, `xsd:decimal`, `xsd:date`. These data types provide more accurate type information compared to the CHAR/NUM designations used in SDTM. Additionally, ISO 8601 formatted variables were assigned to an XSD data type of `xsd:datetime`, `xsd:date`, `xsd:time`, or `xsd:duration` based upon their IG definition.

OWL Class	Usage and Instantiation
Documentation Section	Sections below the IG domain specification are instances of the class `cdiscs:DocumentationSection` and have the following properties: ▪ Section Name ▪ Section Label ▪ Order of the Section after the IG domain specification table Assumptions below the IG domain specification are incoming references to the Documentation Section. Note that the use of sections and section labels is inconsistent within the SDTM IG. If sections have not been explicitly defined for a domain, e.g. CE, then the Section has been assigned the Domain Label, e.g. Clinical Events.
Assumption	Assumptions below the IG domain specification are instances of the `cdiscs:Assumption` class and have the following properties: ▪ Order of the Assumption within an Assumption Section ▪ Assumption Text Assumptions do not have any incoming references.

Table 6. OWL Classes for SDTM IG

4.4 SEND IG 3.0

The SEND IG 3.0 RDF dataset imports the SDTM 1.2 and the SEND Controlled Terminology RDF datasets. The following table describes the classes used by SEND IG 3.0.

OWL Class	Usage and Instantiation
Model	The SEND IG Model is an instance of the `mms:Model` class which is a sub-class of `mms:Context`. Properties have been defined to represent the name, label, and description of the model. Observation Classes are incoming references to the SEND IG Model.
Dataset Context	Observation Classes are instances of `mms:DatasetContext` which is a sub-class of `mms:Context`. The following properties are defined: ▪ Observation Class Name ▪ Observation Class Label ▪ Observation Class Description ▪ Order of the Observation Class within the SEND IG SEND domains within an Observation Class are incoming references to the Observation Class, e.g. EX links to the Interventions Observation Class.

OWL Class	Usage and Instantiation
Dataset	SEND domains are instances of the `mms:Dataset` class which is a sub-class of `mms:Context`. The following properties are defined: ▪ Domain Name ▪ Two-letter Domain Code ▪ Domain Label ▪ Domain Structure ▪ Order of the Domain within the Observation Class Variables within a SEND domain and domain level assumptions are incoming references to a SEND domain.
Column	SEND variables are instances of the class `mms:Column` and have the following properties: ▪ Variable Name ▪ Variable Label ▪ Variable Type as defined in SDTM ▪ Variable Type as defined by an XSD data type ▪ IG Controlled Terms or Format ▪ Codelist URI defined by the NCI EVS ▪ Variable Role ▪ CDISC Notes ▪ CDISC Core Compliance ▪ CDISC Documentation References ▪ Order of the Variable within the Domain ▪ Data Element reference in SDTM 1.x Variable level assumptions specified below the IG domain specification are incoming references to the SEND variable. The model defines the Data Element Types using the more prescriptive XML Schema Definition (XSD) data types including `xsd:string`, `xsd:decimal`, `xsd:date`. These data types provide more accurate type information compared to the CHAR/NUM designations used in SDTM. Additionally, ISO 8601 formatted variables were assigned to an XSD data type of `xsd:datetime`, `xsd:date`, `xsd:time`, or `xsd:duration` based upon their IG definition.
Documentation Section	Sections below the IG domain specification are instances of the class `cdiscs:DocumentationSection` and have the following properties: ▪ Section Name ▪ Section Label ▪ Order of the Section after the IG domain specification table Assumptions below the IG domain specification are incoming references to the Documentation Section. Note that the use of sections and section labels is inconsistent within the SEND IG. If sections have not been explicitly defined for a domain, e.g. DM, then the Section has been assigned the Domain Label, e.g. Demographics.

OWL Class	Usage and Instantiation
Assumption	Assumptions below the IG domain specification are instances of the `cdiscs:Assumption` class and have the following properties: ▪ Order of the Assumption within an Assumption Section ▪ Assumption Text Assumptions do not have any incoming references.

Table 7. OWL Classes for SEND IG

4.5 ADaM 2.1

The ADaM 2.1 RDF dataset imports the standard CDISC schema. The following table describes the classes used by ADaM 2.1.

OWL Class	Usage and Instantiation
Model	The ADaM Model is an instance of the `mms:Model` class which is a sub-class of `mms:Context`. Properties have been defined to represent the name, label, and description of the model. ADaM Data Structures (Subject Level Analysis Dataset, Basic Data Structure, Adverse Event Data Structure) are incoming references to the ADaM Model.
Dataset Context	ADaM Data Structures are instances of `mms:DatasetContext` which is a sub-class of `mms:Context`. The following properties are defined: ▪ Data Structure Name ▪ Data Structure Label ▪ Data Structure Description ▪ Order of the Data Structure within the ADaM document ADaM Data Structures have no incoming references.

Table 8. OWL Classes for ADaM

4.6 ADaM IG 1.0

The ADaM IG 1.0 RDF dataset imports the ADaM 2.1 and the ADaM Controlled Terminology datasets. The following table describes the classes used by the ADaM IG 1.0.

OWL Class	Usage and Instantiation
Model	The ADaM IG Model is an instance of the `mms:Model` class which is a sub-class of `mms:Context`. Properties have been defined to represent the name, label, and description of the model.

OWL Class	Usage and Instantiation
Dataset	ADaM domains are instances of the `mms:Dataset` class which is a sub-class of `mms:Context`. The following properties are defined: ▪ Dataset Name ▪ Dataset Label ▪ Dataset Structure ▪ Dataset Description ▪ Order of the Dataset within the IG Variable Groupings within an ADaM domain are incoming references to an ADaM dataset.
Variable Grouping	Variable Groupings are instances of `mms:VariableGrouping` which is a sub-class of `mms:Context`. The following properties are defined: ▪ Variable Grouping Label ▪ Order of the Variable Grouping within the IG ADaM Variables within a Variable Grouping are incoming reference to that Variable Grouping.
Column	ADaM variables are instances of the class `mms:Column` and have the following properties: ▪ Variable Name ▪ Variable Label ▪ Variable Type as defined in ADaM ▪ Variable Type as defined by an XSD data type ▪ IG Controlled Terms or Format ▪ Codelist URI defined by the NCI EVS ▪ CDISC Notes ▪ CDISC Core Compliance ▪ CDISC Documentation References ▪ Order of the Variable within the Dataset Variables have no incoming references.

Table 9. OWL Classes for ADaM IG

5 A Walk-through of the SDTM IG model in RDF

The previous sections introduced the RDF schemas and the RDF datasets that were created to represent the existing CDISC foundational standards in RDF. This section contains a detailed walk-through of one of the SDTM IG 3.1.3 model. For the walk-through and the screen shots TopBraid Composer Free Edition was used. For more information on RDF toolsets, please consult the companion Reviewer Guide.

5.1 The rdf.cdisc.org Package

After downloading the package you should have the following directory and file structure.

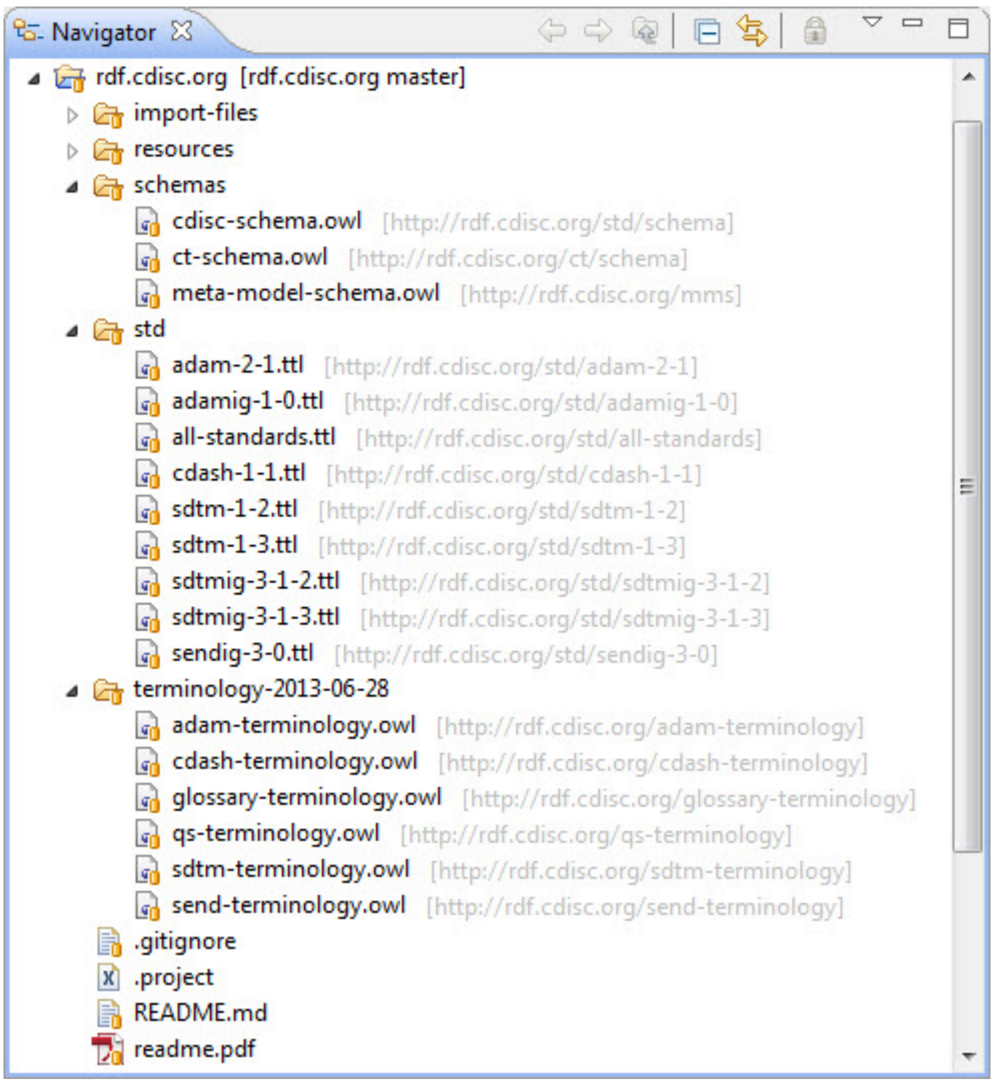


Figure 5. The rdf.cdisc.org Package

The `import-files` folder contains a set of Excel files that were used to do a bulk upload of the models in RDF. They are not needed any further once the RDF datasets are available, but they are included in the package so that users new to RDF have additional means to explore and understand the models.

The `resources` folder contains the SKOS and the Dublin Core RDF files that are imported by the meta-model schema. These vocabularies are used to self-document the schemas.

The `schemas` and `std` folders contain the RDF schemas and RDF datasets described in the previous sections.

The files in the `terminology-yyyy-mm-dd` folder are dated and intended for demonstration purposes only. The latest RDF terminology files can be downloaded from the NCI EVS web site by following the link from `cdisc.org/terminology` and are compatible with the RDF models described in this guide. The dated terminology files can be replaced by the latest ones if needed without breaking the other RDF datasets.

Note. The W3C defines several serialization formats so that an RDF graph can be written to and read from a file. The files with the extension `owl` have been serialized as RDF/XML. The files with the extension `ttl` have been serialized using Turtle. Both are W3C recommendations, but Turtle is gaining more popularity over RDF/XML. The NCI EVS currently publishes the terminology files, including the meta-model schema as RDF/XML files. It is possible that in the future all files in this distribution will be published using the Turtle format. RDF toolsets should have no problem reading either format in a consistent way.

5.2 Exploring RDF Graphs

The intrinsic data model of RDF is a graph data model. For large models, such a graph or its serialization can be daunting. RDF toolsets provide different visualizations to more easily create, navigate, explore, and query RDF graphs. This guide uses TopBraid Composer Free Edition to illustrate these capabilities and give more insight into how the CDISC standards in RDF look like.

Before exploring the SDTM IG 3.1.3 model, the remainder of this section shows a number of representations of the RDF graph data, how the representation relates to the graph data model, and how the RDF graph can be navigated by using a resource URI as a data link. Throughout the examples qualified names are used as a shorthand for the full URI.

The first example in Figure 6. RDF Graph Representation is an actual graph representation of SDTM IG 3.1.3 defining a number of datasets that make up the Trial Design Model.

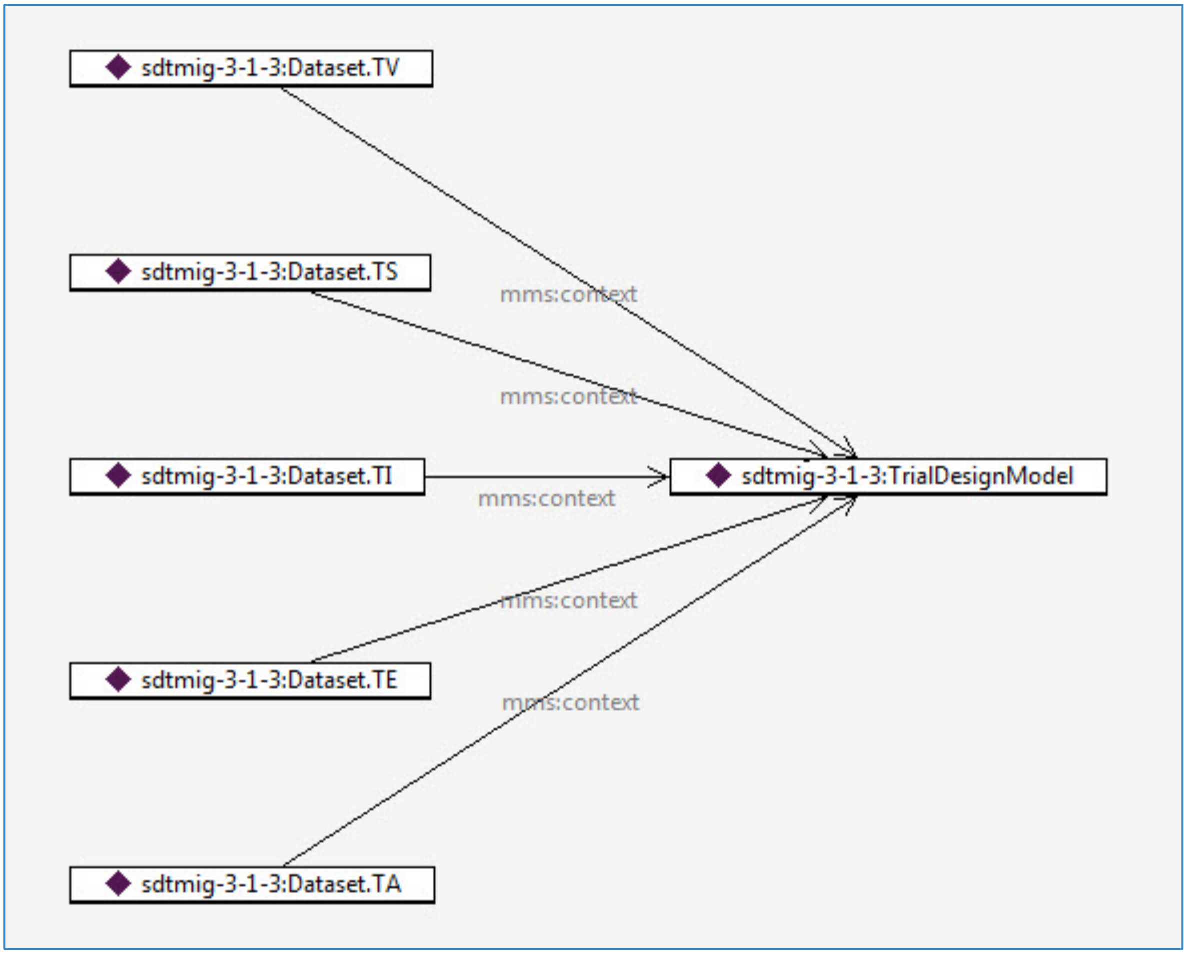


Figure 6. RDF Graph Representation

This graph expresses the following triples:

```
sdtmig-3-1-3:Dataset.TV mms:context sdtmig-3-1-3:TrialDesignModel .
sdtmig-3-1-3:Dataset.TS mms:context sdtmig-3-1-3:TrialDesignModel .
sdtmig-3-1-3:Dataset.TI mms:context sdtmig-3-1-3:TrialDesignModel .
sdtmig-3-1-3:Dataset.TE mms:context sdtmig-3-1-3:TrialDesignModel .
sdtmig-3-1-3:Dataset.TA mms:context sdtmig-3-1-3:TrialDesignModel .
```

The next example in Figure 7. Resource Form of the Trial Design Model gives another view on the resource representing the Trial Design Model. The triples for which this resource is an object are listed as **Incoming References** for the `mms:context` predicate. The triples for which this resource is a subject are listed as **Other Properties**, e.g.

```
sdtmig-3-1-3:TrialDesignModel mms:context sdtmig-3-1-3:Model.SDTMIG-3-1-3 .
sdtmig-3-1-3:TrialDesignModel mms:contextLabel "Trial Design Domain" .
sdtmig-3-1-3:TrialDesignModel mms:ordinal "6" .
```

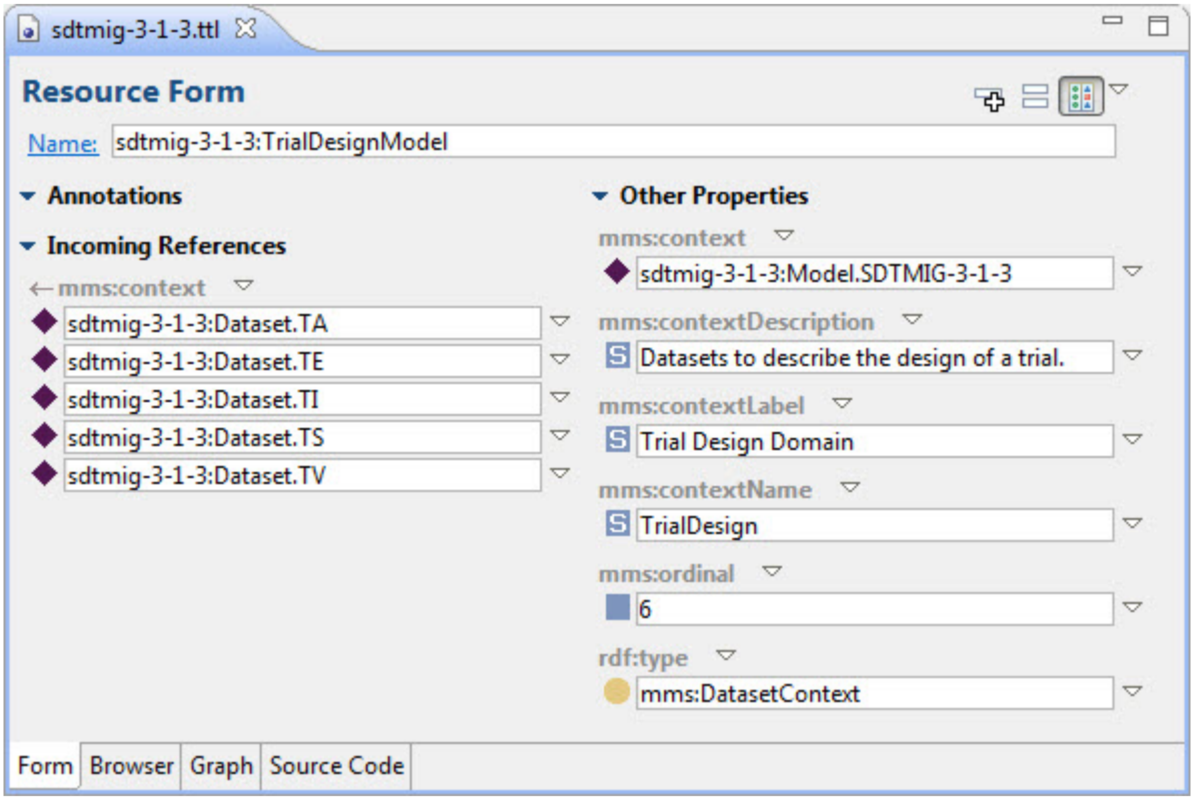


Figure 7. Resource Form of the Trial Design Model

The resource form can also show the full URI of the resource.

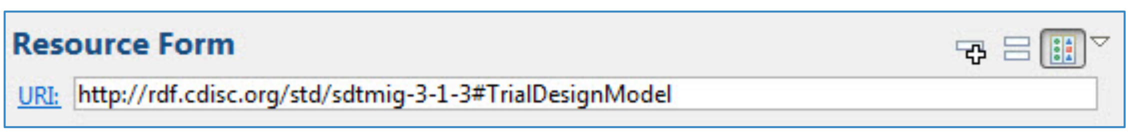


Figure 8. Resource URI

Like a hyperlink in a web page, a resource URI provides an anchor to navigate across resources, which allows a user to "walk the graph" and explore information as linked data.

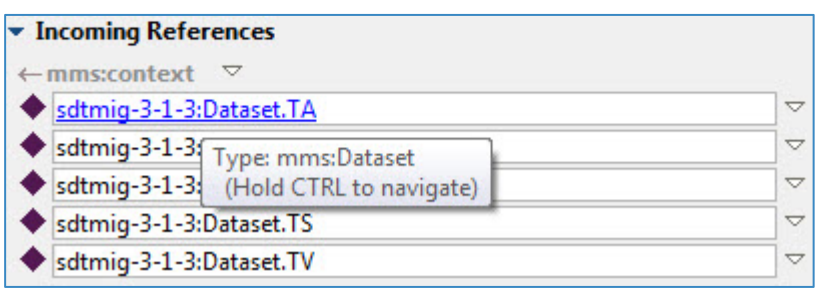


Figure 9. Navigating Across Resources

Following the link takes us to the TA domain.

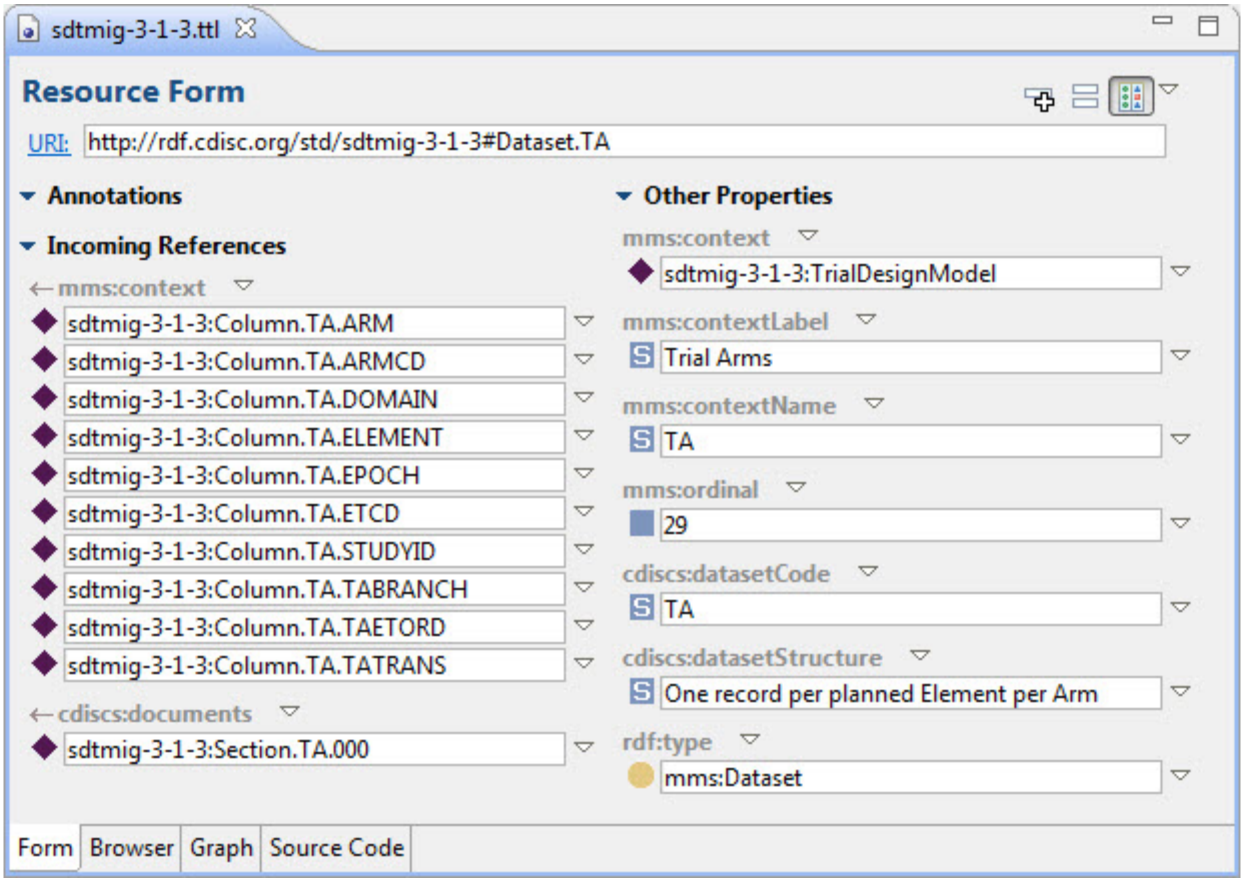


Figure 10. Resource Form of the TA Domain

Another view of the same resource shows a text serialization in Turtle format.

```
sdtmig-3-1-3:Dataset.TA
     a                        mms:Dataset ;
     mms:context              sdtmig-3-1-3:TrialDesignModel ;
     mms:contextLabel         "Trial Arms"^^xsd:string ;
     mms:contextName          "TA"^^xsd:string ;
     mms:ordinal              "29"^^xsd:positiveInteger ;
     cdiscs:datasetCode       "TA"^^xsd:string ;
     cdiscs:datasetStructure  "One record per planned Element per Arm"^^xsd:string .
```

Figure 11. TA Domain in Turtle Format

This text representation shows all the triples of which the TA Domain is the subject. This example also shows that all literals have an explicit XSD data type.

Note. The previous examples have shown different ways to represent and explore resources in an RDF graph. Whatever the actual representation is, it is important to always keep in mind that it is just a view on the RDF resources. The intrinsic model is always that of a graph model defined by a set of triples.

Finally, a very short pointer to SPARQL, a W3C standard that defines a protocol and a graph query language for RDF. Figure 12. Example SPARQL Query shows a query that finds all SDTM variables with codelist C71113 in the SDTM IG 3.1.3 model.

Query Editor | Query Library

```
# Find all SDTM variables with codelist C71113
PREFIX mms: <http://rdf.cdisc.org/mms#>
PREFIX sdtmct: <http://rdf.cdisc.org/sdtm-terminology#>
SELECT ?domain ?variable ?label
WHERE {
  ?dataElement mms:dataElementValueDomain sdtmct:C71113 .
  ?dataElement mms:context/mms:contextLabel ?domain .
  ?dataElement mms:dataElementName  ?variable .
  ?dataElement mms:dataElementLabel ?label .
}
```

Figure 12. Example SPARQL Query

The `WHERE` clause defines a graph pattern using variables (indicated by the question mark) for the data element, the domain, the variable, and the label. Figure 13. Example SPARQL Result Set shows how for every match of the graph pattern, the variables of the graph pattern are bound to a result value and returned as a row in the result set.

[domain]	variable	label
Concomitant Medications	CMDOSFRQ	Dosing Frequency per Interval
Exposure	EXDOSFRQ	Dosing Frequency per Interval
Substance Use	SUDOSFRQ	Use Frequency Per Interval

Figure 13. Example SPARQL Result Set

RDF toolsets like TopBraid Composer provide editors to define, manage, and run SPARQL queries.

5.3 A Walk-through of the SDTM IG 3.1.3 Graph

With the previous examples in mind, this section is going to walk through the SDTM IG 3.1.3 model. In this example the walk-through starts with the top-level context, the model itself, and drills down recursively following the context relationship.

5.3.1 Classes and Instances

Figure 14. SDTM IG Classes and Instance Count shows the class hierarchy of the meta-model and CDISC schemas after being populated with the SDTM IG 3.1.3 model.

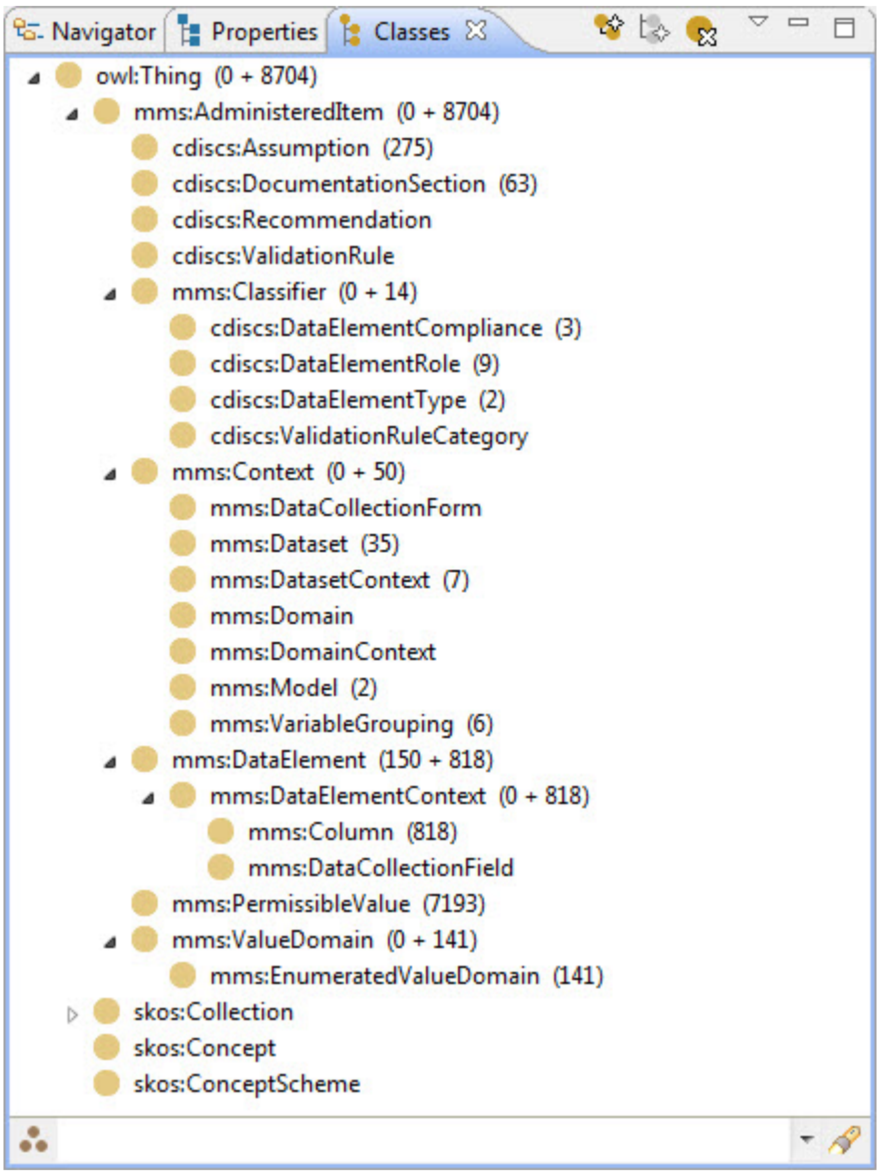


Figure 14. SDTM IG Classes and Instance Count

The instance count shows that there are 8704 resources across the different classes, most of which are permissible values (codelist elements). The Model class has two instances.

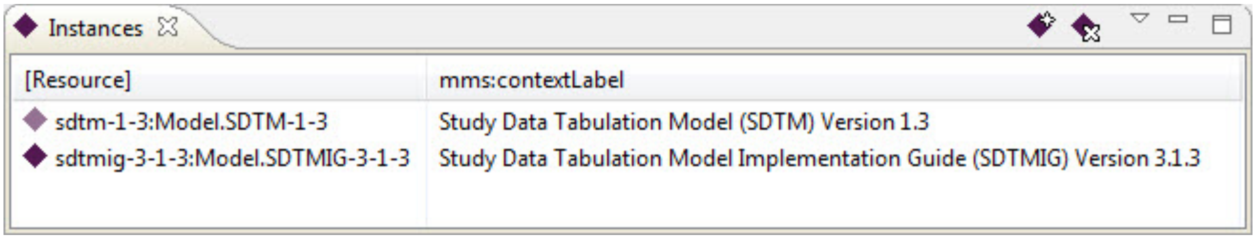


Figure 15. SDTM Model Instances

5.3.2 The SDTM IG 3.1.3 Model

From the Model instances, follow the link to the SDTM IG 3.1.3 Model.

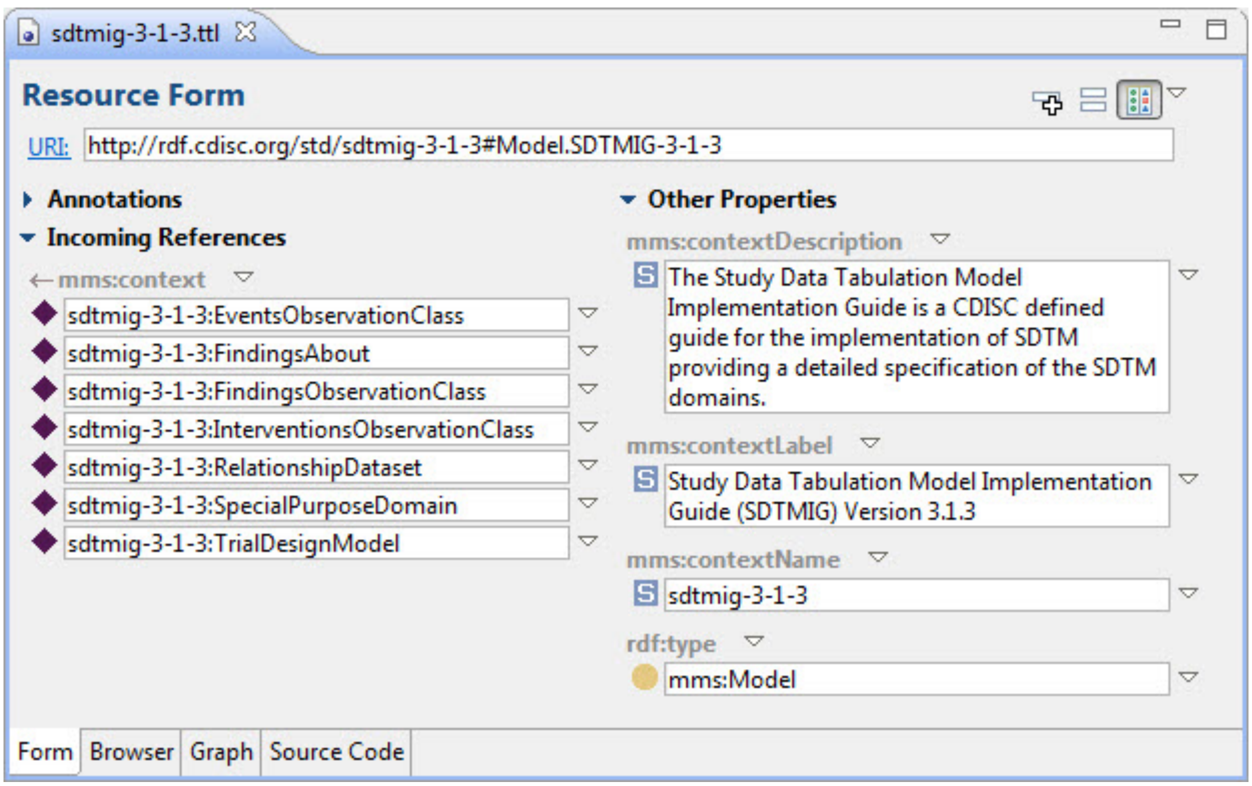


Figure 16. Resource Form of SDTM IG 3.1.3

The different observation classes, the relationship datasets, the special purpose domains, and the Trial Design Model each have a context defined that links to the SDTM IG model resource.

5.3.3 The Events Observation Class

Next, follow the link to the Events Observation Class.

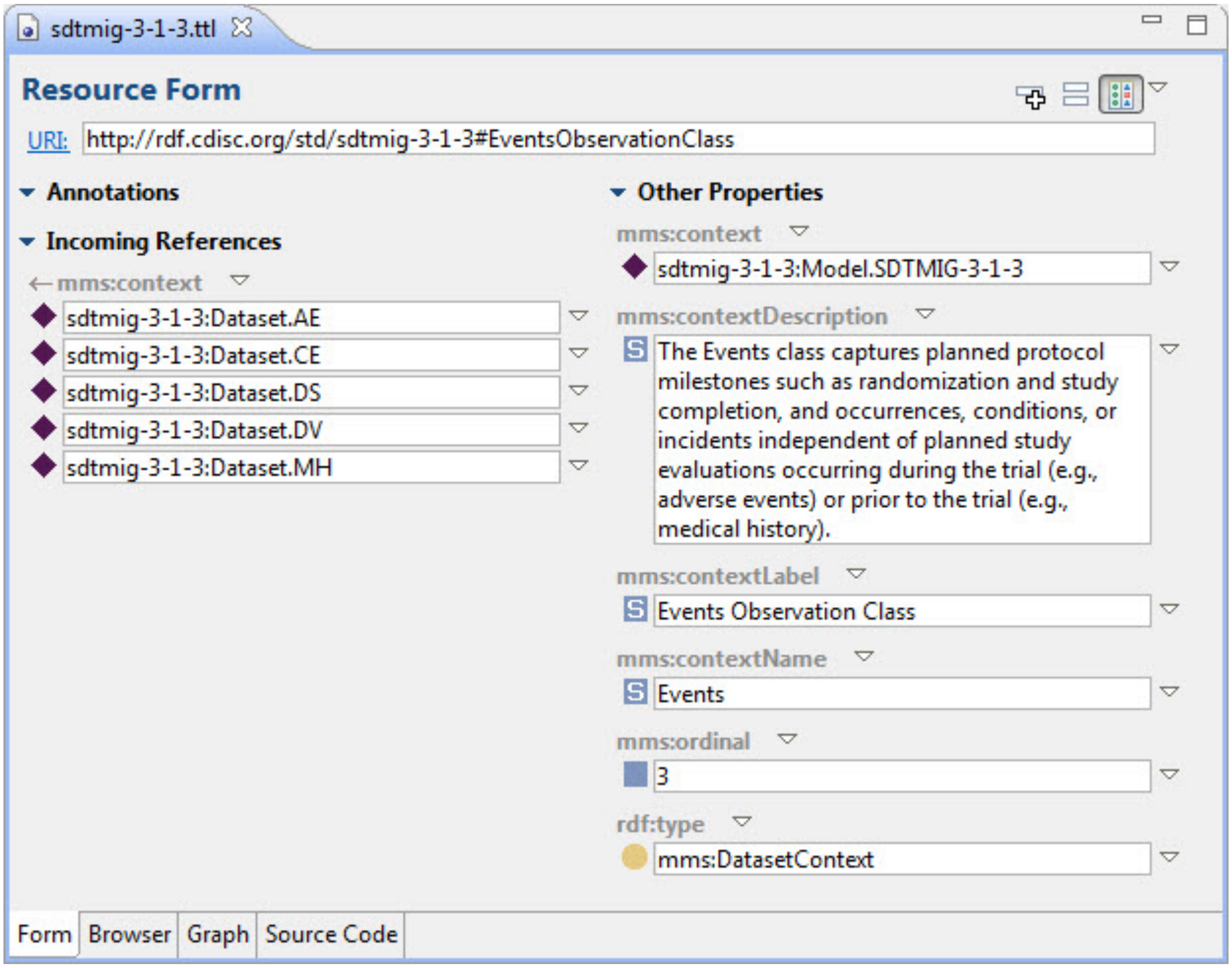


Figure 17. Resource Form of the Events Observation Class

The Events Observation Class is documented with a context description from the Implementation Guide. This shows how easy it is to document every resource in the model, so that all the model elements and the model itself become fully self-documenting.

The Events Observation Class has five SDTM domains (datasets) again related by the context property.

5.3.4 The Adverse Events Domain

Next, follow the link to the SDTM domain Adverse Events.

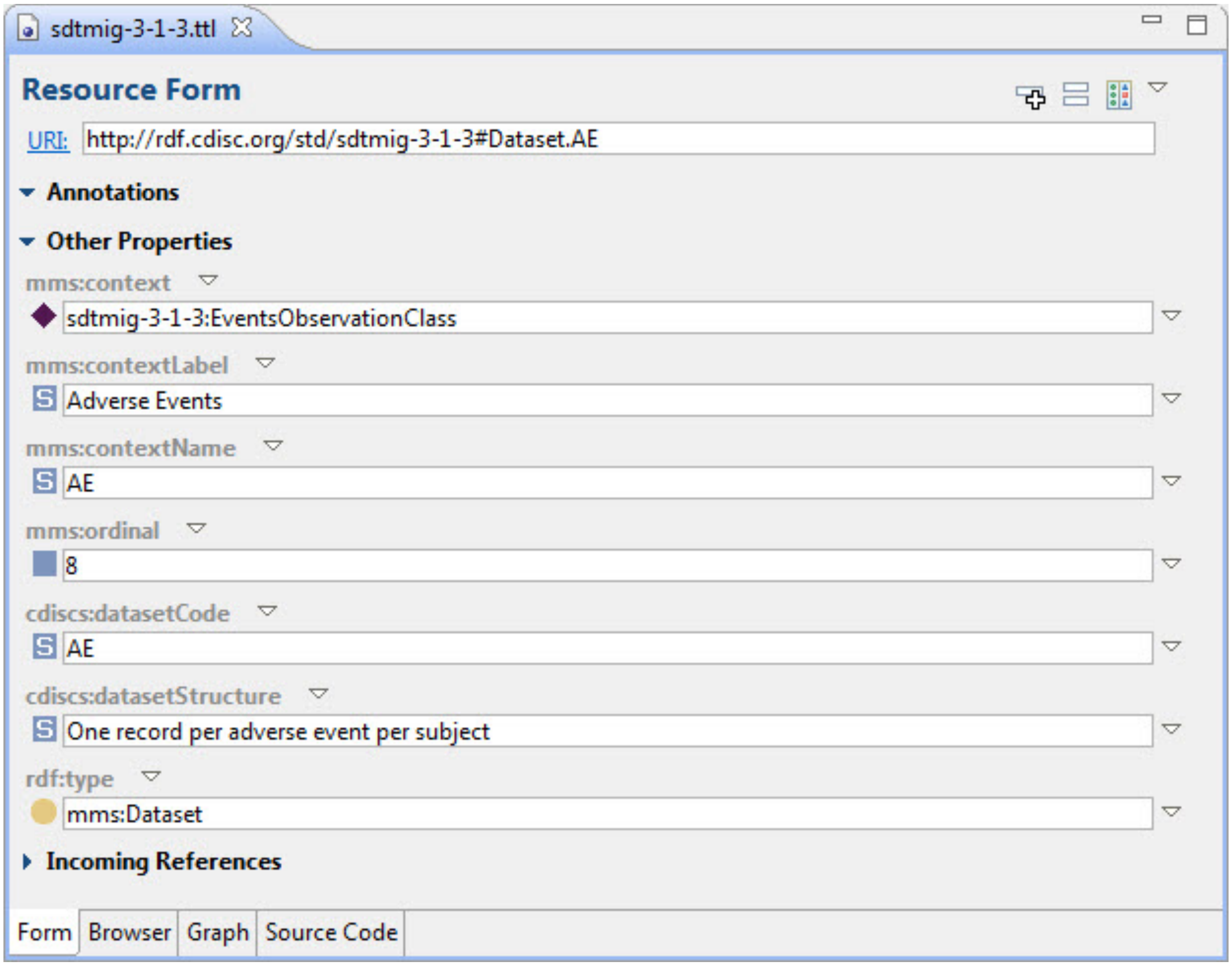


Figure 18. Resource Form of the Adverse Events domain

The Adverse Events domain has only a few properties (label, dataset code, ordinal, and dataset structure) defined by the Implementation Guide.

There is a long list of incoming references for three different predicates (`mms:context`, `cdiscs:about`, and `cdiscs:documents`).

The `context` property relates the AE variables to the AE domain.

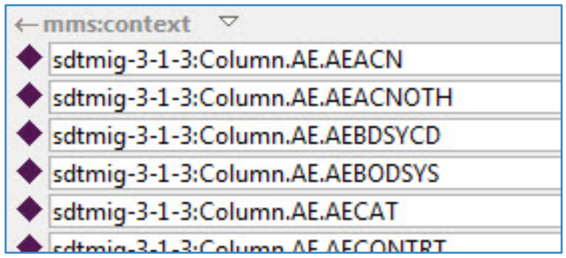


Figure 19. AE Variables of the AE Domain

The `about` property relates all the specific IG assumptions. The URI minting convention uses the domain code, followed by the section number and the assumption number within the section.

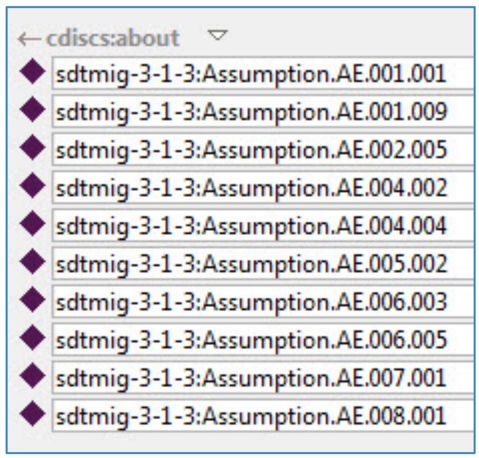


Figure 20. Assumptions About the AE Domain

The `documents` property relates all the IG text sections that follow the AE domain table to the AE domain.

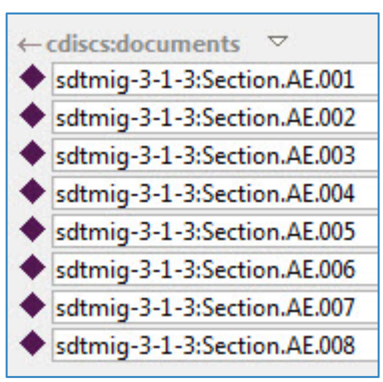


Figure 21. IG Text Sections of the AE Domain

5.3.5 The Variable Action Taken with Study Treatment

Next, follow the link from the AE domain to the variable AEACN.

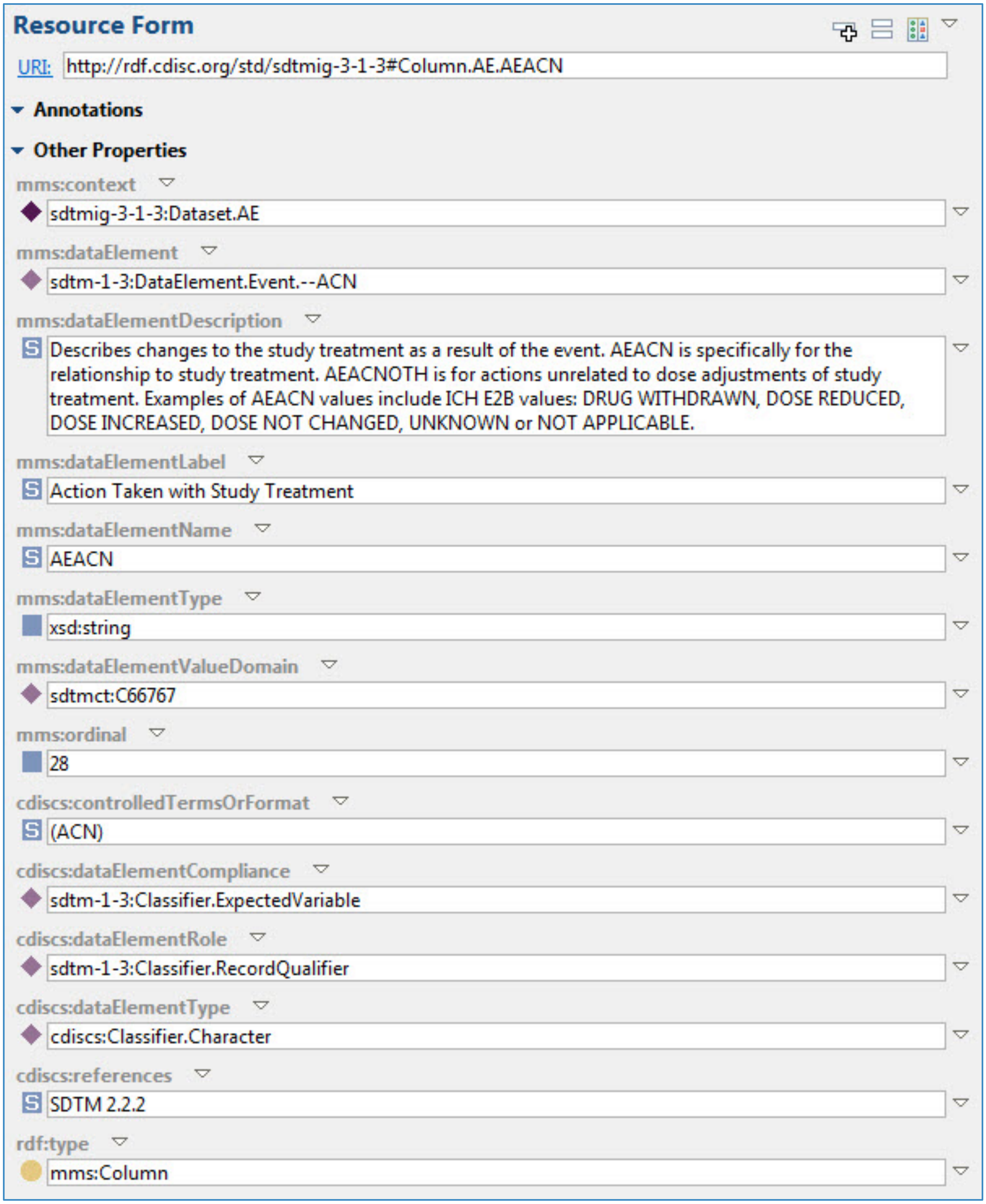


Figure 22. Resource Form of the AEACN Variable

The properties defined in the Implementation Guide are shown. Some of these properties such as name and label are literal values. Other properties have been implemented as resource values, e.g. the fact that AEACN is an expected variable is not indicated by a string literal, rather it is implemented by a property pointing to the `sdtm-1-3:Classifier.ExpectedVariable` resource.

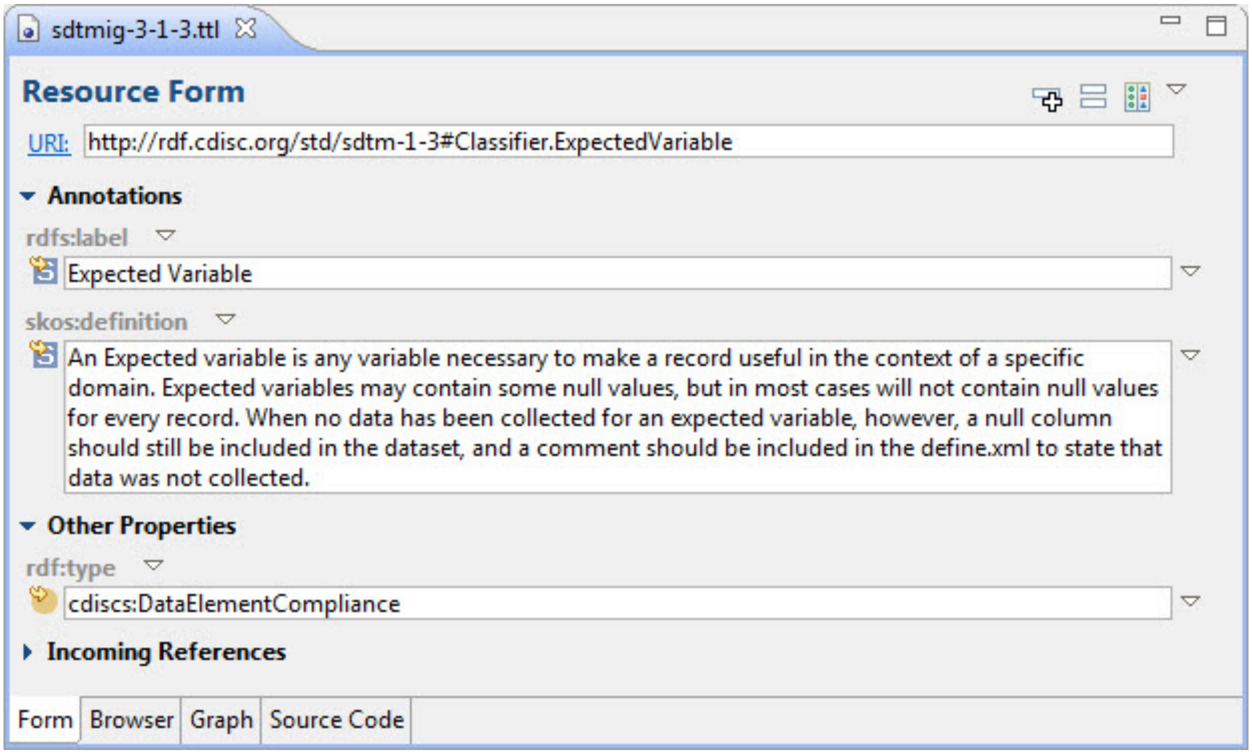


Figure 23. Resource Form of the Expected Variable Classifier

There are several advantages of having the notion of Expected Variable available as a resource. We can provide the resource with additional metadata to make the model fully self-documenting, e.g. in this case we added a SKOS definition with the defining text taken from the SDTM standard. We also know all the expected variables in the model by just looking at the incoming references for the Data Element Compliance predicate.

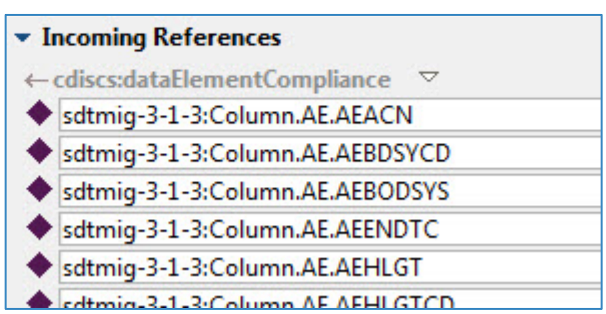


Figure 24. Data Elements with Compliance Expected Variable

Similarly, the AEACN variable is also classified as a Record Qualifier and Character variable. These notions have also been modeled as resources.

There are two more links from the AEACN variable, one to its codelist C66767 and one to its corresponding data element --ACN in the SDTM 1.3 model.

5.3.6 The Codelist C66767

Next, follow the link from the SDTM variable AEACN to its codelist C66767.

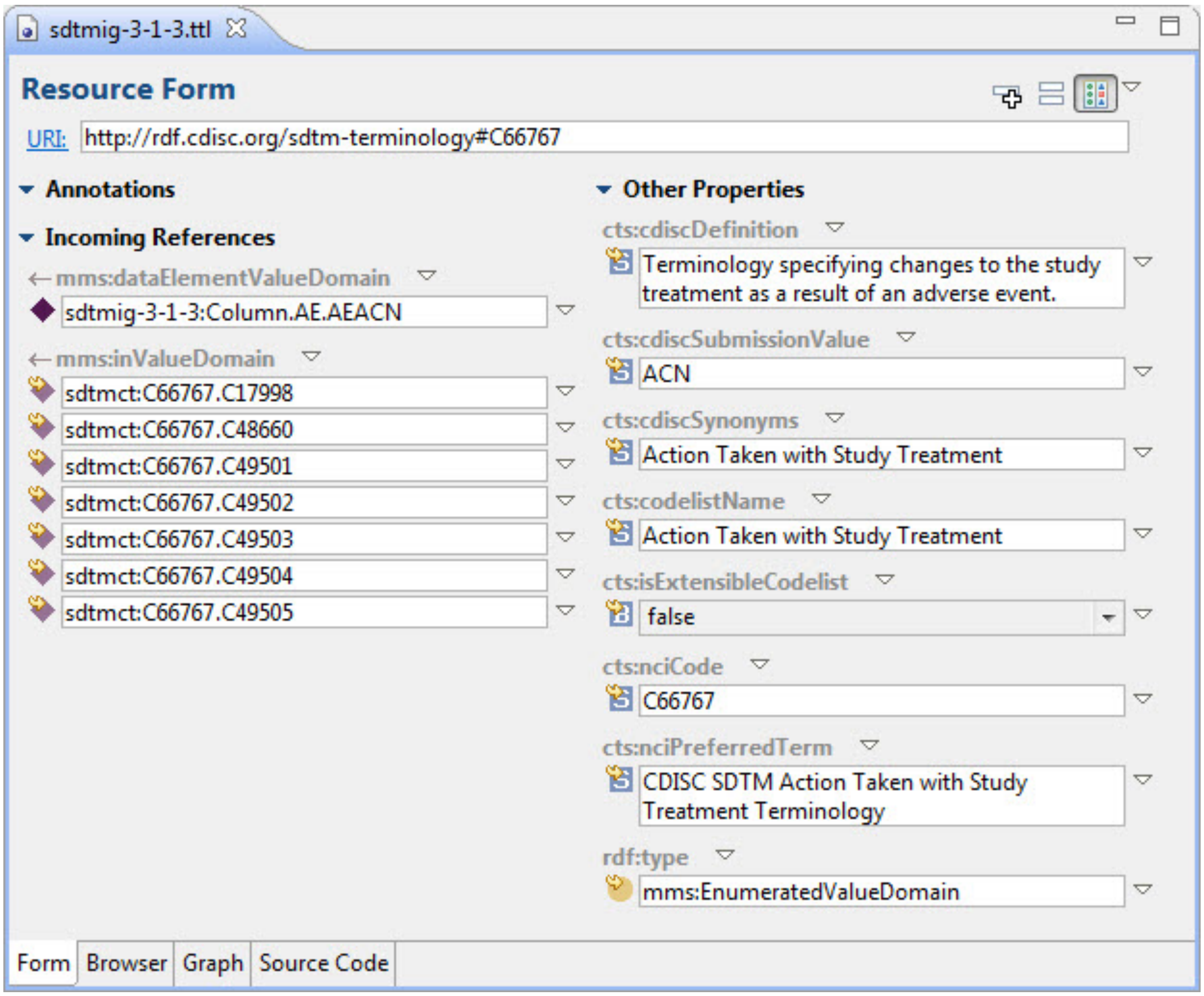


Figure 25. Resource Form of the Codelist C66767

The resource form shows the codelist attributes as published by the NCI EVS, e.g. this codelist is not extensible. The incoming references from the predicate `mms:inValueDomain` refer to the codelist elements.

Next, follow the link to the codelist element C49501.

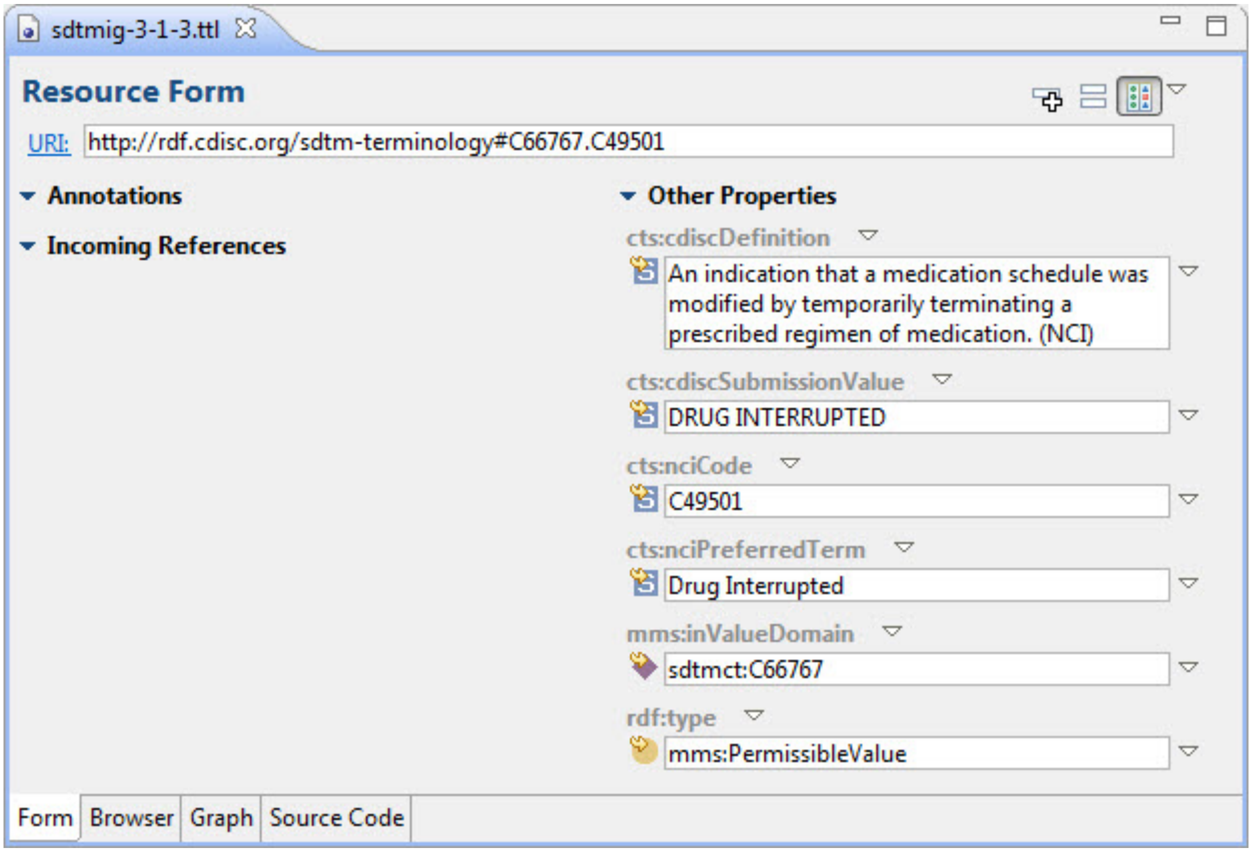


Figure 26. Resource Form of the Codelist Element C49501

The resource form shows the codelist element attributes as published by the NCI EVS, e.g. the definition, submission value, and preferred term. There are no further incoming references.

5.3.7 The SDTM 1.3 Data Element --ACN

Before, the link from the AEACN variable to its codelist C66767 was followed. It is also possible to follow another link from AEACN to its corresponding data element in the SDTM 1.3 model. Following this link leads seamlessly from the current SDTM 3.1.3 model defined in the `sdtmig-3-1-3.ttl` file to the SDTM 1.3 model defined in the `sdtm-1-3.ttl` file.

SDTM 1.3 defines generic data elements that guide the creation of new SDTM domains. From the example in Figure 27. Resource Form of the --ACN Data Element one can see that the use of SDTM 1.3 data elements promotes reuse in the IG domains, in particular if both models can be linked as shown.

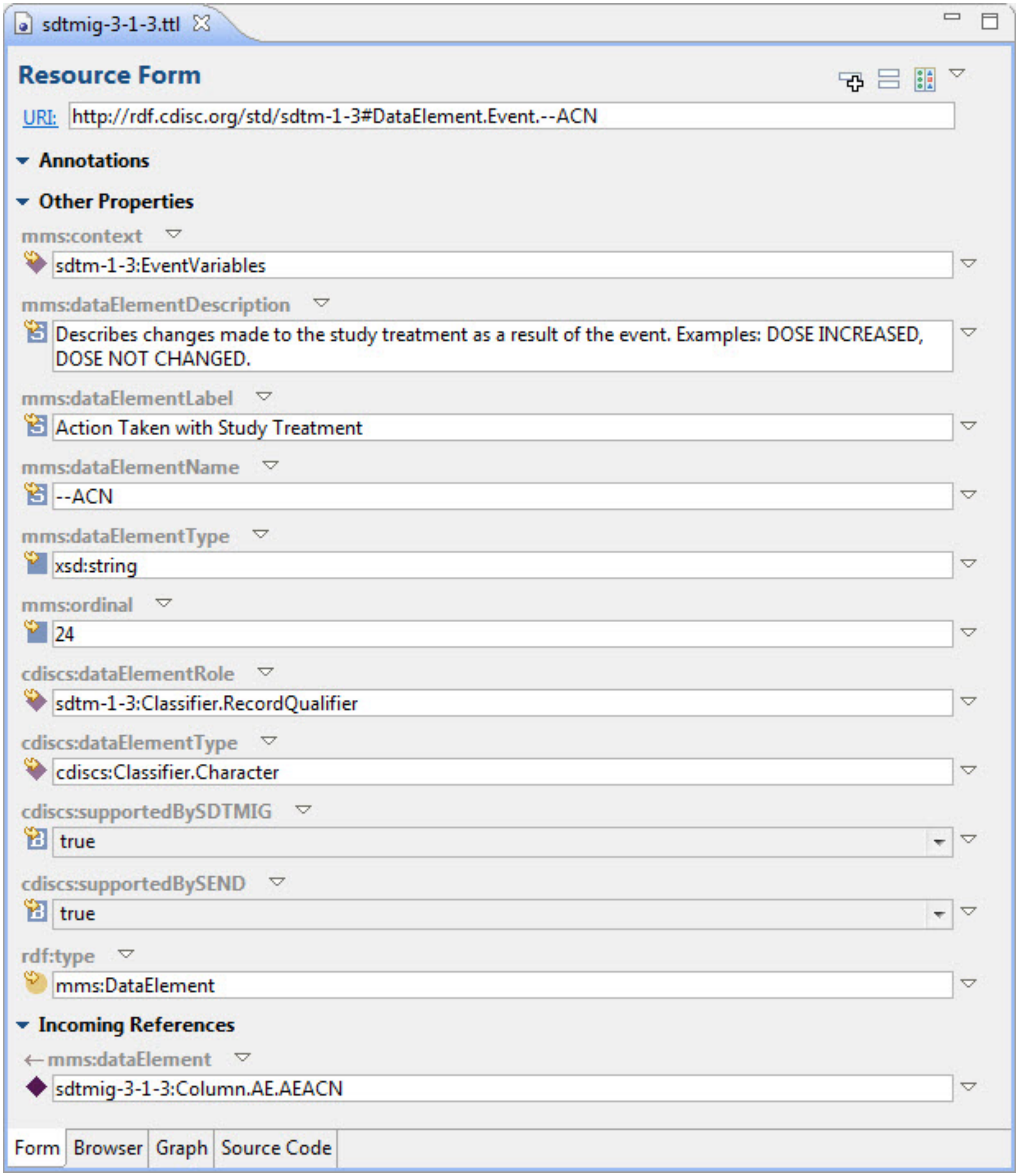


Figure 27. Resource Form of the --ACN Data Element

The RDF models consistently define an additional and more precise XSD data type for each data element. The original character/numeric classifier and the CDISC Controlled Terms and Format moniker have been kept, but a separate XSD data type was added using the predicate `mms:dataElementType`. In particular, the ISO 8601 data types defined in IG are described by the more precise `xsd:date`, `xsd:time`, `xsd:datetime`, and `xsd:duration` types as shown in the following example.

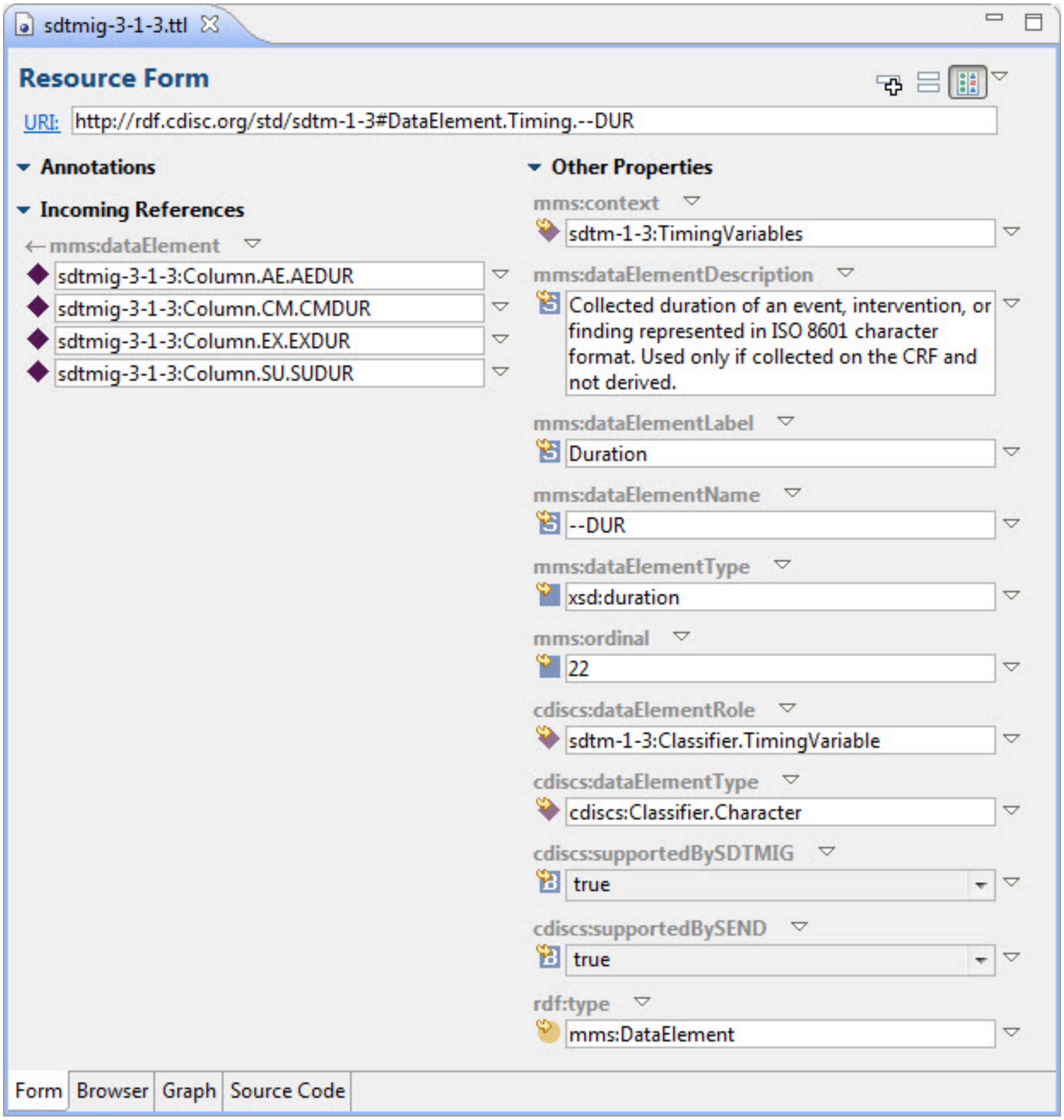


Figure 28. Example of ISO 8601 Data Types

5.3.8 Assumptions

Previously the link from the AE domain to the variable AEACN was followed. The resource form for the AE domain also had incoming links for the assumptions and the assumption sections related to the AE domain.

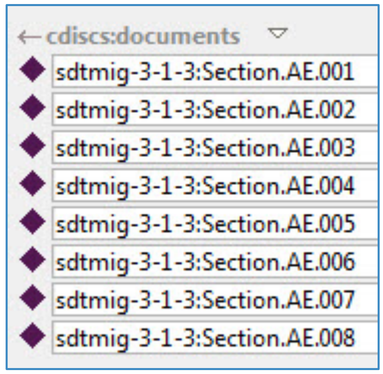


Figure 29. Assumption Sections for the AE Domain

Next, follow the link for AE Section 4.

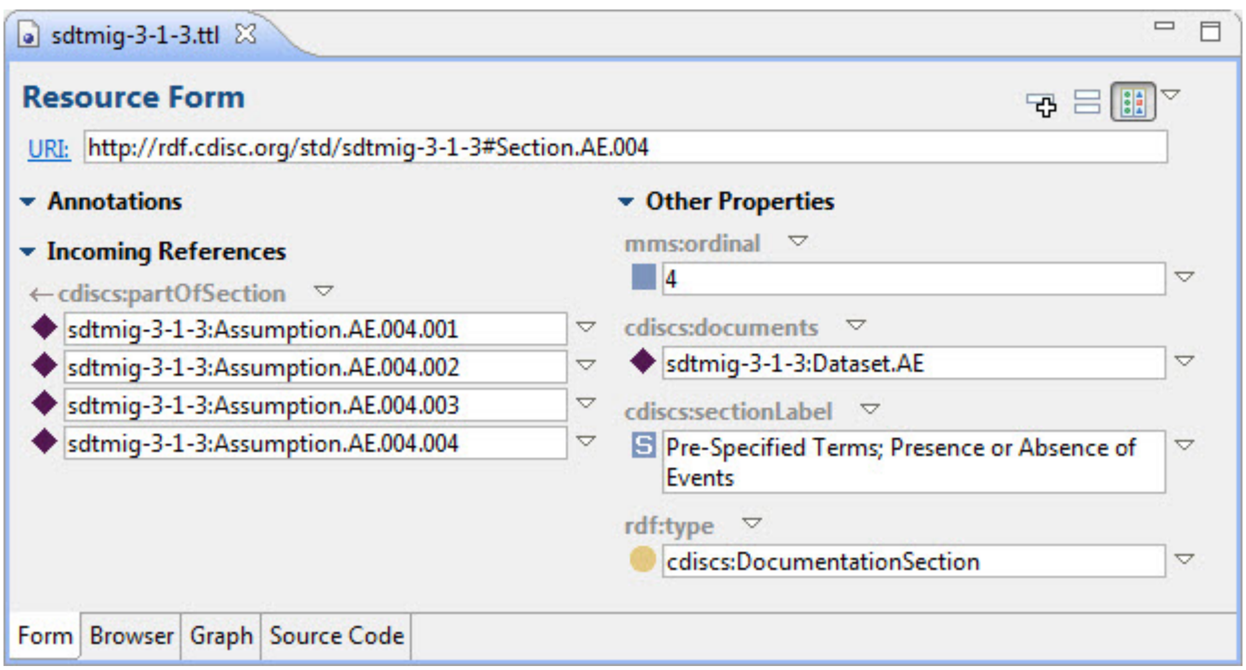


Figure 30. Resource Form for AE Assumption Section 4

The resource form shows that there are four assumptions in this section.

Next, follow the link for the second assumption in AE Section 4.

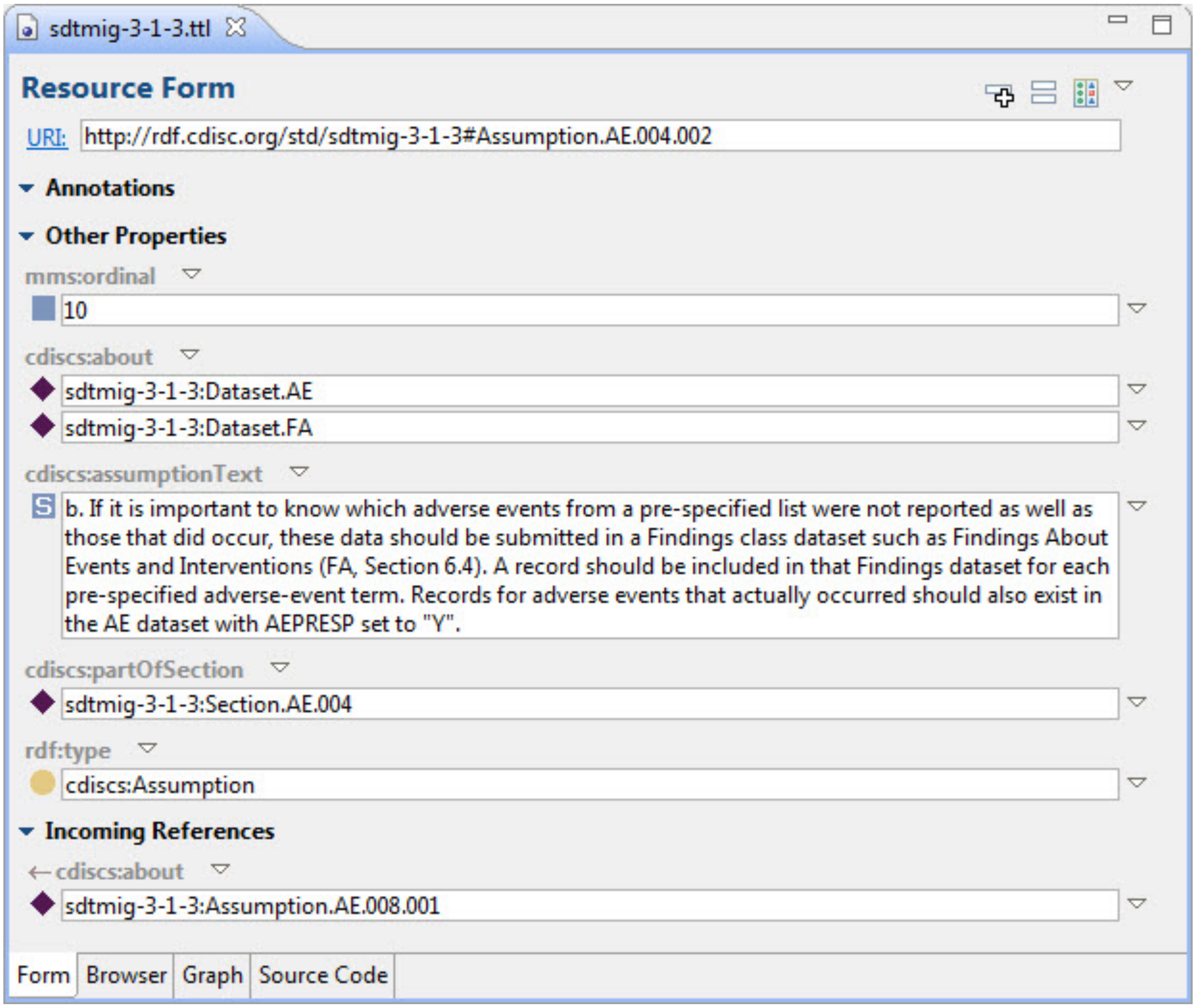


Figure 31. Resource Form for AE Section 4 Assumption 2

The assumption is about the AE and FA domains. There is also another assumption in AE section 8 that talks about this assumption.

Appendix: Representations and Warranties, Limitations of Liability, and Disclaimers

CDISC Patent Disclaimers

It is possible that implementation of and compliance with this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any claim or of any patent rights in connection therewith. CDISC, including the CDISC Board of Directors, shall not be responsible for identifying patent claims for which a license may be required in order to implement this standard or for conducting inquiries into the legal validity or scope of those patents or patent claims that are brought to its attention.

Representations and Warranties

"CDISC grants open public use of this User Guide (or Final Standards) under CDISC's copyright."

Each Participant in the development of this standard shall be deemed to represent, warrant, and covenant, at the time of a Contribution by such Participant (or by its Representative), that to the best of its knowledge and ability: (a) it holds or has the right to grant all relevant licenses to any of its Contributions in all jurisdictions or territories in which it holds relevant intellectual property rights; (b) there are no limits to the Participant's ability to make the grants, acknowledgments, and agreements herein; and (c) the Contribution does not subject any Contribution, Draft Standard, Final Standard, or implementations thereof, in whole or in part, to licensing obligations with additional restrictions or requirements inconsistent with those set forth in this Policy, or that would require any such Contribution, Final Standard, or implementation, in whole or in part, to be either: (i) disclosed or distributed in source code form; (ii) licensed for the purpose of making derivative works (other than as set forth in Section 4.2 of the CDISC Intellectual Property Policy ("the Policy")); or (iii) distributed at no charge, except as set forth in Sections 3, 5.1, and 4.2 of the Policy. If a Participant has knowledge that a Contribution made by any Participant or any other party may subject any Contribution, Draft Standard, Final Standard, or implementation, in whole or in part, to one or more of the licensing obligations listed in Section 9.3, such Participant shall give prompt notice of the same to the CDISC President who shall promptly notify all Participants.

No Other Warranties/Disclaimers. ALL PARTICIPANTS ACKNOWLEDGE THAT, EXCEPT AS PROVIDED UNDER SECTION 9.3 OF THE CDISC INTELLECTUAL PROPERTY POLICY, ALL DRAFT STANDARDS AND FINAL STANDARDS, AND ALL CONTRIBUTIONS TO FINAL STANDARDS AND DRAFT STANDARDS, ARE