Ontological Representation of the SDTM AE Domain

The CDISC Pilot AE data is now represented fully in the Study Ontology. The first 9 records of the AE domain have been entered manually in the knowledge base. The team is requested to review this information and comment. Below are the suggested approaches to review AE data in the Ontology. The steps go from the least to most technical.

# Round-Tripping

Does the data extracted from the knowledge base match the source data?

A fundamental question is does the data coming out match the going in.

The source data is located on GitHub at /CTDasRDF/source/AE\_imputed.xlsx

The output data is located on GitHub at /CTDasRDF/source/AE\_roundtrip.xlsx

Verify that the two match

Recall that additional dummy data were added to the AE source domain to enable testing of all the variables in the AE domain. These imputed data are shown in red in the AE\_imputed.xlsx file.

## AE Modeling in the Ontology

Does the modeling of AE data in the ontology make sense? Are there alternate approaches that we should consider?

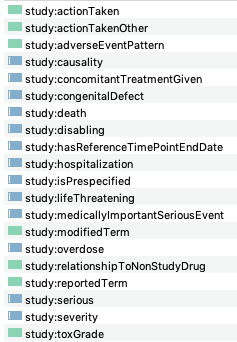
1. Open study.ttl
2. Expand the study:Entity class and examine the subclasses
3. Review the definitions of each class, stored as ?s skos:definition <definition>
4. Examine the properties of the study:AdverseEvent class? Do they make sense? Are there alternate suggestions?
   1. The properties can be identified using the following SPARQL query:

**SELECT** ?p

**WHERE** {

?p rdfs:domain study:AdverseEvent .

}

* 1. The properties are:  
     

(These are in addition to the properties available to all StudyComponent classes and Event classes)

# Examine the mappings from the Ontology to the SDTM AE domain

1. Open sdtm.ttl
2. Open the sdtm:SDTMComponent class and drill down to sdtm:SDTM\_AE.ttl
3. Examine mappings (expressed in SPARQL) from the ontology to SDTM AE  
   these are found as SPIN rules and look like this

# AEDECOD and Full MedDRA mapping

**CONSTRUCT** {

**?this** sdtm:hasAELLT ?AELLT .

**?this** sdtm:hasAELLTCD ?AELLTCD .

**?this** sdtm:hasAEDECOD ?AEDECOD .

**?this** sdtm:hasAEPTCD ?AEPTCD .

**?this** sdtm:hasAEHLT ?AEHLT .

**?this** sdtm:hasAEHLTCD ?AEHLTCD .

**?this** sdtm:hasAEHLGT ?AEHLGT .

**?this** sdtm:hasAEHLGTCD ?AEHLGTCD .

**?this** sdtm:hasAESOC ?AESOC .

**?this** sdtm:hasAESOCCD ?AESOCCD .

**?this** sdtm:hasAEBODSYS ?AEBODSYS .

**?this** sdtm:hasAEBDSYCD ?AEBDSYCD .

}

**WHERE** {

**?this** a sdtm:SDTMRecord .

**?this** sdtm:hasEntity ?subject .

?subject study:afflictedBy ?ae .

**?this** sdtm:hasEvent ?ae .

?ae a study:AdverseEvent .

?ae code:hasCode ?LLT .

?LLT skos:prefLabel ?AELLT .

?LLT code:hasIdentifier ?AELLTCD .

?LLT mdra:hasPT ?pt .

?pt skos:prefLabel ?AEDECOD .

?pt code:hasIdentifier ?AEPTCD .

?pt mdra:hasHLT ?hlt .

?hlt skos:prefLabel ?AEHLT .

?hlt code:hasIdentifier ?AEHLTCD .

?hlt mdra:hasHLGT ?hlgt .

?hlgt skos:prefLabel ?AEHLGT .

?hlgt code:hasIdentifier ?AEHLGTCD .

?hlgt mdra:hasSOC ?soc .

?soc skos:prefLabel ?AESOC .

?soc code:hasIdentifier ?AESOCCD .

?soc skos:prefLabel ?AEBODSYS .

?soc code:hasIdentifier ?AEBDSYCD .

}

# Generate a tabular AE domain

1. Open the sdtm-cdisc01.ttl file
2. Run Inferences to infer derived data, e.g. full MedDRA hierarchy and others.
3. Open the sdtm:SDTMDomain class. there you will find an embedded SPARQ query to generate the AE\_roundtrip tabular domain.
4. Execute the Query on this file.

# Optional

Although out of scope for the pilot, you’ll find some SHACL constraints. This is a preliminary attempt to demonstrate the advantages of RDF in the ability to link validation rules to the data in a machine-processable format. They are located in the study.ttl file. e.g. linked to the AdverseEvent class.

Comments are welcome.