Report to the Hyperledger Climate Action SIG Standards WG

The ontology

- Currently hosted [here](#)
  - Source: [wiki](#) and [protobufs](#)
- Find it on [GitHub](#)
- Find it on [Triply](#)
  - [SPARQL endpoint](#) (until 6/02/2023)
- Find a nice(r) UI [here](#)

Notes
- Edits made are indicated by comments containing the word EDIT
- Problems/TODOs indicated by comments containing the word FIX
- Definitions that are not final are indicated by `rdfs:comment` tags containing the words NOT FINAL

What's next?

- Complete definitions/axioms

Some considerations:
- `Control` class: Procedures, Plans, Objectives/Goals, Processes, Requirements, Specifications, standards all Controls?
- `Role subClassOf Agent`? (makesClaim and verifies properties)
- From [dcterms](#)
  - [http://purl.org/dc/terms/MethodOfAccrual](#) for Claim?
  - [http://purl.org/dc/terms/Policy](#) subClassOf Control
  - [http://purl.org/dc/terms/dateAccepted](#) to show acceptance of a verified claim
- Which properties are compulsory and must be specified? (Protobuf should have content in all fields)

Vocabularies

Who?	GitHub	[Triply](#)	Hosted	TODO
CDM	[CDM](#) (OWL)	[CDM](#) (Turtle)	[http://purl.org/aiaontology/cdmvocabulary](#)	Add terms from [glossary](#); add properties from aia
Gold Standard	[GS](#) (OWL)	[GS](#) (Turtle)	[http://purl.org/ai](#)	Add properties

			aontology/gsvocabulary	from aia
VCS	VCS (OWL)	VCS (Turtle)	http://purl.org/aiaontology/vcsvocabulary	Largely incomplete

What's next?

- How to approach vocabularies? As an "addendum" to aia, or independently?
- Host on actual website, no redirect.
- SPARQL endpoint
- Content negotiation

POC Annotated project

- This is a semantic annotation of this CDM Project.
- Proposed workflow (newspaper article approach)

What's next?

- Data validation
- Webscraping or other similar semi-automatic way to extract data.
- Tool to enable users to semantically annotate their data (e.g. PDDs) and export RDF.

Useful resources

- Turtle Parser: Takes RDF turtle data as input and parses it into RDF triples.
  - Dereferenceable blank nodes
- Developed as part of an effort to develop an RDF data validator:
  - Potentially unfeasible because all the data must be categorised into classes and every individual must have a `rdf:type` property, which is often not the case.
  - It can be possible if you have the end goal of validating the data in mind while writing the data as well as the ontology. Very detailed `disjointWith` declarations needed, for example.
  - The program would additionally need the capability to detect inverses and other relations in the object properties (which this program cannot yet do).
  - Checking subject and object against domain and range.
- Use with a tool like this (downloads ontologies locally to check declarations; saves time)

- Webscraper for CDM project page. Not complete (indicated with comments).
  - Gold Standard tags seem inconsistent

Other:
- Protégé for viewing/editing ontologies (note some bugs; use in conjunction with IDE of your choice and extensions like RDFox and RDF Sketch)
- Triply for storing and querying linked databases
- PyLODE to generate OWL documentation (the easiest way of the using the tool is the python module which only requires 3 lines of code to produce the documentation)
- Python libraries for RDF related development:
  - RDFLib
  - PyLDAPI
  - Ontology rdflib python (youtube)
- Linked Data API training
- Linked Data APIs
- Ontotext Metadata Studio
- Open Semantic Search

Less useful resources

- INCEpTION for semantic annotation (huge text files, can't export triples)
- LinkedSDG extracts key concepts related to sustainable development from text documents; only generates a sort of word cloud and no linked data. Technical description here.
- Stardog: useful for storing very large amounts of data; knowledge graphs. Less ideal for creating ontologies and linked data
- Apache Tomcat: hard to work with, little documentation. Triply works better for our purposes.