















Index of /file/work/OntologySummit2013/Hackathon-Clinics/HC-05_Ontology-of-OntologyEvaluation/owl

Name	Last modified	Size	Description
 Parent Directory		-	
 application.ttl	02-Apr-2013 02:21	315	
 catalog-v001.xml	02-Apr-2013 02:21	4.9K	
 evaluation.ttl	02-Apr-2013 02:21	1.9K	
 evaluation_method.ttl	02-Apr-2013 02:21	508	
 evaluation_process.ttl	02-Apr-2013 02:21	3.9K	
 lifecycle.ttl	02-Apr-2013 02:21	580	
 metrics_model.ttl	02-Apr-2013 02:21	1.3K	
 ontology.ttl	02-Apr-2013 02:21	3.4K	
 ontology_characteristics.ttl	02-Apr-2013 02:21	2.6K	
 ontology_characteristics_ogware.ttl	02-Apr-2013 02:21	4.0K	
 ontology_requirements.ttl	02-Apr-2013 02:21	2.4K	
 ontology_usage.ttl	02-Apr-2013 02:21	1.5K	
 organizational_process_and_maturity_models.ttl	02-Apr-2013 02:21	3.2K	

OWL 2 Ontology of Ontology Evaluation

This directory contains an OWL 2 representation of the Ontology of Ontology Evaluation which is intended to be faithful to the conceptual drawings developed during the HC-05 hackathon.

A [Turtle](#) file corresponds to each of the drawings. [evaluation.ttl](#) imports all of the other files.

These files were tested using Protege 4.2.0 (Build 295), available at protege.stanford.edu. Each is self-contained (resulting in some duplicate class definitions, which isn't a problem). All the files use the same http://ontology.cim3.net/file/work/OntologySummit2013/Hackathon-Clinics/HC-05_Ontology-of-OntologyEvaluation/owl/evaluation namespace, which is resolvable modulo HTTP content negotiation.

Below we discuss some of the decisions made and issues encountered in developing this OWL 2 representation. Many are generally applicable to other representations. Some of the files also contain `TODO` comments.

Associating Value Types with Properties

We make heavy use of `owl:allValuesFrom` restrictions (OWL's version of FOL `forall`), which are essentially class-specific versions of `rdfs:range`. Many of our properties are reused (a good thing), so `rdfs:range` is generally too restrictive.

We should verify that such restrictions are appropriate, both within and potentially outside the ontology. For instance, agents can potentially manage many things (ontology requirements, people, companies, etc.). In some cases, it may be more appropriate to use the inverse of the property, e.g. an ontology requirement is managed by an agent.

Associating Multiple Value Types with Properties

In some cases, a class uses the same property (e.g. defines) with values of multiple types. We generally model this using an `owl:allValuesFrom` restriction to an anonymous `owl:unionOf` class. The disadvantage of this approach is that `owl:unionOf` closes off its list of classes (e.g. additional value types can't be added elsewhere). Again, we should ensure that such restrictions are appropriate, both within and potentially outside the ontology. It may be necessary to merge lists or use less constrained restrictions.

`owl:someValuesFrom` restrictions (OWL's version of FOL `exists`) can be used to indicate that a property has some (but not necessarily all) values of a given type, but it also implies `owl:minCardinality 1`, which may not be appropriate in many cases.

Loosely Associating Properties with Classes

It's often useful (e.g. to human readers or GUIs) to indicate that a given property is relevant to a class even if we're not able to say anything more about it.

A tautological `owl:minQualifiedCardinality 0` restriction can be used to associate a property relating 2 classes without XXX. Qualified cardinality restrictions were added in OWL 2.

Similarly, a tautological `owl:minCardinality 0` statement can be used to associate a property with a class without specifying the type of its value.

Cardinalities

The conceptual diagrams do not specify cardinalities, so we don't otherwise specify them. Cardinalities should presumably be specified, where appropriate, as the ontology is refined.

Ontology Characteristics

TBD classes vs. instances. punning. restrictions.

[Mike Dean](#), 1 April 2013