Semantics II

Matthew J. Graham
CD^3

Methods of Computational Science
Caltech, 2015 April 9

matthew graham

concept schemes

- controlled vocabulary: a closed list of terms that can be used for classification
- taxonomy: a controlled vocabulary with hierarchy (term - sub-term)
- thesaurus: a taxonomy with broader/narrower term, synonymous term, top term, scope note and related term
- subject heading list, terminology, glossary, faceted classification

simple knowledge organization system

- W3C standard for expressing knowledge concept schemes in a machine-understandable way
- Set of RDF properties and RDFS classes to express the content and structure of a concept scheme as an RDF graph:

- Collection
- ConceptScheme
- prefLabel
- definition
- broader
- CollectableProperty
- Concept
- altLabel
- example
- narrower
- OrderedCollection
- hiddenLabel
- scopeNote
- related

skos example

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xml:base="http://www.astro.physik.uni-goettingen.de/~hessman/rdf/IVOAT">
    <skos:ConceptScheme rdf:about="">
        <skos:hasTopConcept rdf:resource="#temperatureScales"/>
    </skos:ConceptScheme>
    <skos:Concept rdf:about="#absoluteTemperatureScale">
        <skos:prefLabel>absolute temperature scale</skos:prefLabel>
        <skos:definition>absolute temperature scale</skos:definition>
        <skos:broader rdf:resource="#temperatureScales"/>
        <skos:narrower rdf:resource="#Kelvin"/>
        <skos:related rdf:resource="#celsiusTemperatureScale"/>
        <skos:related rdf:resource="#temperature"/>
    </skos:Concept>
</rdf:RDF>
```

sparql protocol and rdf query language

- W3C standard for RDF query language and data access protocol
- SELECT, CONSTRUCT, ASK, DESCRIBE
- WHERE, FILTER, OPTIONAL, UNION
- ORDER BY, DISTINCT, REDUCED, LIMIT, OFFSET

sparql example

```
PREFIX table: <http://www.example.org/PeriodicTable#>
SELECT ?name ?symbol ?weight ?number ?color
FROM <http://www.example.org/PeriodicTable.owl>
WHERE
{
    ?uranium table:name "uranium".
    ?uranium table:atomicWeight ?uraniumWeight.
    ?element table:name ?name.
    ?element table:symbol ?symbol.
    ?element table:atomicWeight ?weight.
    ?element table:atomicNumber ?number.
    OPTIONAL { ?element table:color ?color }.
    FILTER ( ?weight > ?uraniumWeight )
}
ORDER BY ASC[?weight]
```

ontologies

An ontology is a formal specification of a shared conceptualization

-- it is a data model that represents a set of concepts within a domain and the relationships between those concepts

Ontologies generally describe:

- Individuals: the basic or "ground level" objects
- Classes: sets, collections, or types of objects
- Attributes: properties, features, characteristics, or parameters that objects can have and share
- Relations: ways that objects can be related to one another
- Events: the changing of attributes or relations

web ontology language (owl)

- W3C standard for authoring ontologies
- Based on RDF (OWL semantically extends RDFS)
- Regarded as one of the fundamental technologies underpinning the Semantic Web
- OWL allows descriptions of:
  - relations between classes (e.g. disjointness)
  - cardinality (e.g. "exactly one")
  - characteristics of properties (e.g. symmetry)
  - enumerated classes

owl components

- Data is interpreted as:
  - a set of **individuals**
  - a set of **property assertions** relating the individuals to each other
  - a set of **axioms** placing constraints on sets of individuals (**classes**) and the types of relationships allowed between them

- For example, the *family* ontology:
  - "hasMother" is only present between two individuals when "hasParent" is also present
  - members of "HasTypeOBlood" are never related via "hasParent" to members of "HasTypeABBlood"
  - If Ada "hasMother" Anne and Ada is "HasTypeOBlood" then Anne is not "HasTypeABBlood"

inference

Computers provide reasoning services over a knowledge domain where the domain and the knowledge have been formally and rigorously specified and reasoning algorithms have been implemented in a way which that computer can apply. For example:

- Sheep only eat grass
- Grass is a plant
- Plants and parts of plants are disjoint from animals and parts of animals
- Vegetarians only eat things which are not animals or parts of animals

=> sheep are vegetarians!

inference examples

- relevant terms for stage:

  start some (succeeded_by some stage1) and end some (preceded_by some stage2)

- related terms for term:

  part_of some class1 or develops_from some class1 or has_part from some class1

- lowest common ancestor:

  superclassOf(class1 or class2)

software

- Jena (jena.sourceforge.net)
- Protege (protege.stanford.edu)
- cwm (www.w3.org/2000/10/swap/doc/cwm.html)