Functional Requirements for Vocabulary and Information model registry systems

Stephen Richard 2015-08-18 22:43

Vocabulary registry functions:

Scope is a concept vocabulary—each item represents a unique concept; the definition of the item in natural language, along with examples, images etc. convey the intention of the concept in a manner that should make it clear to all within the target community for the vocabulary.

Target applications:

- reference for communities to document the meaning of terminology they adopt
- Annotation of resources (keywords)
- Semantic search—concept expansion, synonomy
- Terminological-property domain validation in data
- Pick lists for user interfaces

Functions:

- Basic CRUD operations on concept; registry needs to allow various policies on update and deletion (i.e. no delete, only deprecate)
- Concepts have URI, prefLabel, altLabels, definition, source (minimum)
- Resolve URI to definition, source and labels for human consumption
- Language localization?
- Status properties for concepts, e.g. proposed, adopted, deprecated, superseded (with links to successor).
- CRUD operations on Relations/objectProperties (add new relationships). Relationships/associations are first order concepts
- Get related concepts;
    - simplest case are SKOS type relations;
    - general relation navigation (CRUD operations on relations, essentially becomes an ontology?),
    - transitive relations/transitive closure,
    - relation hierarchies (e.g. ‘chapter’ is a kind of ‘partOf' relation).
- Get URI associated with Label
- Find concepts related to text strings (full text search—labels + definitions)
- CRUD operations for mapping relations between concepts in different collections or conceptSchemes
- Define collections of concepts, e.g. as a domain for some property value;
- CRUD operations on collections and collection items;
- formal concept of a ‘conceptScheme’ that has properties like intended scope (concept space), isCovering, isComplete, uniqueConcepts, hierarchical…

- Given concept and collection, report if concept is member of collection (for domain validation processes)
- Get all members of a collection (tools use to construct pick lists…)
- Track source of concept (citation), steward for concept (who put in, who is responsible for maintenance and status)
- auto-complete functionality
- OpenRefine resolution services

Information model registry

Scope is formal representation of data types, which would include a definition of the 'types' or 'entities', and specification of a collection of attributes, with domains and cardinalities for those attributes, constituting the representation of instances of that type/entity. These could be implemented as JSON objects, XML elements, rows in a relation, RDF graphs etc, all different representations of the same fundamental type. Note that this kind of registry seems to be the objective of the RDA Data Type Registry work group.

Target applications:

- Reference for communities to document the meaning of entities and attributes in data that they share.
- Discover existing data type and attribute definitions for use in constructing data models, to foster interoperability.
- Machine-assisted data integration, based on identification of matching or 'integratable' attribute content.
- Validation of data instances against a type definition.
- Tools that spin up a UI for a particular data type.

Information concepts:

- 'Integratable' is a somewhat tortured adjective, used here to mean 'capable of integration', i.e. values from different sources can be used in an application as if they were part of a single data set to obtain scientifically sound results. The data integration process might involve transformations such as conversion of measurement units that do not inherently change the meaning of the measurement. See discussion of property values, below.
- Entities, Attributes, and Properties are concepts, thus inherit properties from concepts as defined for vocabulary (above)
- Attribute is a logical implementation for representing a Property value. A given property, e.g. temperature, may be represented in various ways, e.g. as a number (Celsius, Farenheit), or a term (high, medium, low). Attributes representing the same Property should be integratable.
- Property values may be measured or reported using a variety of different methods, e.g. measured with mercury thermometer, alcohol thermometer, infrared sensor, reported as 'Average temperature (over some interval)', 'Peak temperature (over some interval)', 'instantaneous temperature'. These all relate to a general 'temperature' concept, but may not be integratable, and for the purposes of data integration these distinctions need to be documented. There is a continuum from the most general concept of temperature (least likely to be integratable) to a property representing a temperature measurement by a particular observer using a single measurement and reporting method (most likely to be integratable)

- Entity and Type. CSDGM metadata data defines entities and attributes, but this entity concept is essentially a type definition. Type is concerned with the definition of a data structure; any entity has an inherent type, so the concepts are closely related. Thus this text mostly uses both terms like this 'entity/type' (subject to getting a better idea…)

Functions:

- Get property definition
- Get all attributes related to a property
- Get related properties
- Get schema for entity/type. Possible representations (content negotiation?): rdf schema, xml schema, JSON schema, others…
- Get entities that include an attribute
- Get entities that include a property
- Get related entities/types. Allow inheritance of attributes from parent to child types.
- Register new type
    - Create through forms
    - Ingest schema document (XML schema, rdf schema, JSON schema, ISO19110 feature catalog, others…)
- Assert relationships between entities, properties, attributes (similar to general relation functions in vocabulary; basically constructing an ontology ….)
- Find transformation between attributes
- Find similar entities (compare attributes)

Summary

Two kinds of registries: one for vocabulary, one for information model. The Information Model Registry builds on the Vocabulary Registry, because the entity/type, attribute, and property concepts that are its basis are all concepts, thus manageable using the vocabulary registry. The information model registry requires a variety of more complex relationships, functions, and additional attributes on the concepts. For instance an attribute will need a data type, and value domain. An attribute in the context of a type/entity will need a cardinality, and perhaps restriction on value domain. Properties will be a complex vocabulary rooted in abstract 'phenomenon' concepts like temperature, density, length, with more granular properties defined based on context (observer?, environment, intention), reporting method, and measurement method. Transformation methods linking attributes will also be useful. Processes for finding transformations between attributes, determining similarity between entity/type definitions, and ingesting new schema will also be needed.