Validation: requirements and approaches

Dave Reynolds , Epimorphics Ltd, 30 June 2013

Having worked with many parts of the UK public sector on the publication and exploitation of Linked Data, we can identify several areas where validation or related functionality would be beneficial. We briefly summarize these requirements and sketch some of the solution parameters.

A common theme across these is that the requirement is not just to provide for validation of a data set but to provide convenient ways to declare and discover the structure of a data set to guide discovery, presentation and data update.

We see (divergent) requirements to describe, discover and validate structure in the context of:

- regularly structured Linked Data
- vocabulary profiles
- code lists

Regularly structured data

A large fraction of the data published by the public sector is what we might term "regularly structured", the sort of information that is naturally suited to representation in spreadsheets. This includes statistical data, performance metrics, monitoring data (such as environment measurements) and financial information (budgets, outturns, payments).

The requirement here is to define the dimensional structure of the data, the values being measured and associated interpretational metadata (e.g. units of measure). This need has been largely met by the RDF Data Cube vocabulary[1] - originally developed under sponsorship from the UK Government, RDF Data Cube is now a W3C Candidate Recommendation.

The RDF Data Cube provides the notion of a Data Structure Definition (DSD) which is a declarative statement of the structure of the data cube (in terms of dimensions, measures and attributes). A predicate is provided to associate a data set with its DSD. This satisfies the requirements to discover data based on its structure and to understand the structure of the data set containing a given sample (*Observation*).

However, thanks to the open world nature of RDF there was little in the way of closure constraints and what it meant to conform to a DSD was insufficiently clear.

The W3C Government Linked Data working group addressed this issue by defining the notion of a *well-formed RDF Data Cube*. This provides a closed world definition of what it means for a data set to conform to a DSD. It comprises a set of integrity constraints, each of which is defined via a SPARQL ASK query.

Comments

1. To first approximation it is plausible that the RDF Data Cube now provides sufficient capability to declare and validate the Linked Data representations of regularly structured data. Further standards

[1] http://www.w3.org/TR/vocab-data-cube/

work in this area could wait until further experience has been gained with the RDF Data Cube specification, and in particular the integrity constraints.

2. SPARQL was technically sufficient to express all of the integrity constraints needed in this domain. However, some of the constraints[2] were difficult to express and led to unnecessarily expensive queries. Alternative implementation approaches (or special case SPARQL optimizations) will be necessary for practical scale validators.

Vocabulary profiles

Next we look at the issues for data sets that do not follow the cube pattern, for example sets of entities (organizations, locations, facilities) and the relationships between them

The existing standards and best practice enable organizations to publish the vocabularies used in such data in the form of RDFS or OWL ontologies. To declare that a given data set uses some set of ontologies then `void:vocabulary` is often used.

However, in practice there are a number of limitations of this approach:

1. Consumers of such data have no guarantees over what data to expect. Even when an ontology declares cardinality axioms then the instance data may apparently violate those "constraints" thanks to the open world assumption.

2. Data is often published using a "mix and match" approach selecting terms from many ontologies. This reuse of existing terminology aids interoperability but makes it hard to provide tooling to automatically construct data entry or presentation interfaces. Just because a vocabulary is used and declared via a `void:vocabulary` gives no guidance on what parts of the vocabulary are actually used in the data.

3. The complexity of OWL semantics make it a barrier to entry for many publishers in this domain. Users who are already familiar with data modelling using other approaches (whether relational or object oriented) find the terminology and semantics of OWL impenetrable. Specialist training can help bridge this gap but still the differences between a schema-like approach and logic-based approach continue to trip people up.

This suggests there is a need for a schema-like *vocabulary profile* mechanism.

Comments

1. The use cases are not simply validation in the sense of mechanically checking data, but include data set discovery and generation of user interfaces for both presentation and data entry. This strongly favours a declarative approach over validation-by-SPARQL query.[3]

2. In terms of expressivity a subset of OWL (comprising RDFS plus class restrictions, and maybe inverseOf) would be sufficient, when interpreted with closed world semantics (c.f. Clark & Parsia's ISV). However, to make this accessible to the target users then it needs to be expressed in a more

[2] In particular IC-12, duplicate detection.

[3] SPARQL provides an implementation option, and could be used to express additional integrity constraints on top of a schema-like core, but SPARQL queries are not sufficiently inspectable to meet the full set of use cases.

digestible form, ideally a simpler surface syntax and a standalone presentation of the semantics that does not require an understanding of OWL.

3. To support discovery there should be a standard predicate to link an endpoint or data to the profile definition.

Code lists

A broad range of public sector data makes extensive use of controlled code lists to ensure that data can be interpreted, compared and combined. Such controlled lists evolve over time and are typically maintained by standards groups separate from the data modellers and publishers. A common requirement is to be able to declare and validate that a data set references only terms from a particular version of such a controlled list.

Semantically this may seem straightforward. However, operationally the separation between data providers, vocabulary definers and code list maintainers introduces significant issues.

The typical approach to code list management is to define some form of *registry*. A registry comprises a number of *registers* each of which acts a controlled list. Arbitrary complexity is possible in how items within registers and registers themselves may be related and grouped, however the essence is that a register acts a unit of governance. Some management authority defines that all and only those items within a register R are approved for a given use. If it is not in the register it is not valid.

There are several standards for such registries[4] and ongoing work to define a compatible notion of a Linked Data registry[5] and associated vocabulary.[6]

Comments

1. The details of registry definition and operation are outside the scope of the workshop.

2. However, there is a need to be able to define data set structure and validation by reference to such external *services* (as opposed to class declarations or enumerations within the ontology).

Acknowledgements

These notes are partially based on discussions within the UKGovLD working group, in particular with Paul Davidson and Jeremy Tandy.

[4] Such as ISO19135, originating from the geo-spatial community.
[5] https://github.com/der/ukl-registry-poc/wiki
[6] https://raw.github.com/wiki/der/ukl-registry-poc/images/registry-diagram.png