Querying the Registry with RDF

Norman Gray
VOTech/AstroGrid
University of Leicester, UK
(and University of Glasgow, UK)

IVOA Interop, Moscow, 2006 September 18–20

overview

If you convert the Registry to RDF, what do you get?

- Yet another vaguely SQL-like query language.
- All the implications of a given registry entry.
- Shareable overlaid structure (bookmarks, recommendations, saved queries).

rdf quick intro

- All the world is *triples*, consisting of *resources* named by URIs (`http:...` or `ivo:...` or `urn:example#Norman`)
- . . . which have *properties* whose *values* are resources or literals.
- RDF/RDFS/OWL describe these using `rdf:type`, `rdfs:subClassOf`, `owl:symmetricProperty`, and so on.

There is an analogy with XML Schemas, *but it is a loose one* – they're not addressing the same problem. Same for O-O.

rdf/owl/semweb/sql/xml – respective strengths

RDF/OWL/reasoning now largely stable (though The Semantic Web will forever be Vision). Now engineering rather than CS.

Using the architectural principles which let HTML take over the internet. Very open and flexible; has existing powerful query language. Did I mention standards?

RDB to XML to RDF – spectrum of strengths. XML is more natural than RDF where the information density is high, and the information regular or highly constrained; RDF/SW is natural for incomplete or ragged data.

rdf schemas give you reasoning

1. X is SecondaryEducationContentLevel
2. SecondaryEducationContentLevel is a subclass of SchoolContentLevel
3. thus X is SchoolContentLevel

1. Y (CurationDescription) `publisher` X
2. `publisher` hasInverse `publishes`
3. thus X `publishes` Y

Add transitive properties, functional properties, symmetric properties, subclass/subproperty relations, and you magnify what you say.

It’s not the query language that’s the win, here, but the fact that the reasoner can expand the set of assertions in your knowledgebase, by drawing all possible conclusions.

Plus you can add derived types.

what does this look like for the registry?

- RDF Schema versions of XSchema standards for SIA-v1.0, VODataService-v1.0, TabularDB-v0.3, VORegistry-v1.0, and VOResource-v1.0. VOResource written by hand (more idiomatic), the others generated automatically.
- XSLT transformations of XSchema instances of RegistryInterface-0.1, VODataService-v0.5, SIA-0.7, VOResource-0.10, and TabularDB-v0.3 XSchemas, to those v1.0 RDF Schemas (not bug-free, yet).
- Seems to work, but it needs more playing with, and clear use-cases.

rm in rdf: demo

[demo]

The SemWeb payoff in this case is a powerful and supple query language, plus expanded set of types.

benefits

- Retrieve implications that were not explicit (eg, SchoolContentLevel)
- Query using personal or community-specified types (eg, SchoolAtlases), that were not originally present.
- Share those types, which can be lists or ‘dynamic queries’.

Value-added registry browser.

Use-cases, anyone?