Data Service Discoverability

A Metadata Perspective

Alex Ball

University of Bath

Data Service Discoverability Workshop, Santorini, 21–22 April 2016

This is quite a nebulous topic to be dealing with, so I reacted in the way I guess most of you did and thought about Definitions

- A *service* is a capability provided by someone else.
- A *research data service* is one that involves research data in some way…

so to answer one of the questions we were sent, a Web service is one you can interact with over HTTP, a research data service involves research data, so they are not the same but most research data services will also be Web services.

We can imagine varieties across different dimensions:

- Is the service aimed at *people* (👤) or *tools* (⚙)?
- Does the service take *metadata or queries* (🏷) or *data* (🗄) as input?
- Does the service give *metadata* (🏷) or *data* (🗄) as output?

Bearing in mind all the time that A single provider may offer more than one service.

I realise this is probably a much broader understanding of research data services than you were thinking about. From the list of questions we were sent, it looks like you were only thinking about data-in, data-out services. That's valid and sensible, but I wonder if people might expect you to consider more than that. Here are some examples of the richness of services that might fall under this umbrella. ¶

Data description registry or data discovery service

- human-facing GUI for search/browse
- APIs for automated queries, e.g. OAI-PMH, Atom
- query in, metadata out

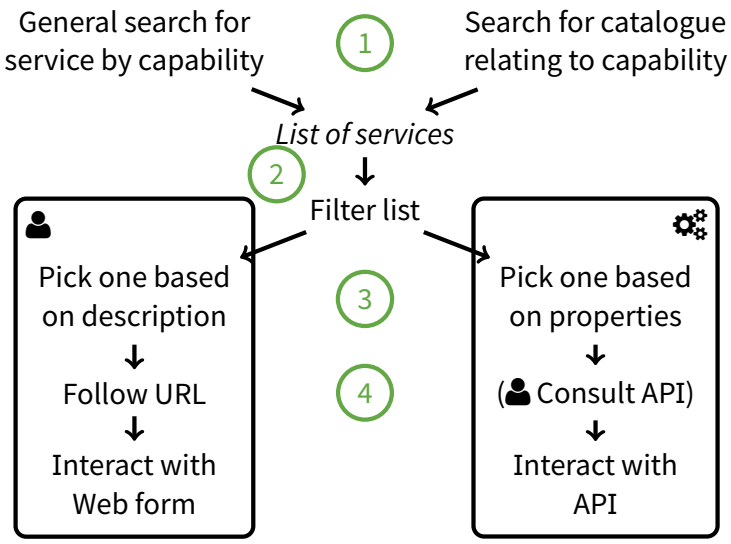


Figure 1: Discovery pathways.

Data repository (ingest)

- human-facing deposit form
- some repositories also support deposit by API, e.g. SWORD
- data and metadata in; metadata out (landing page, DOI)

Data repository (access)

- human-facing GUI for search/browse
- APIs for automated queries, e.g. OAI-PMH, Atom
- query in, metadata/data out

Online data analysis library

- API for submitting data
- raw data in, processed data out

What the user needs to know in order to select and use the service will be different in each case. For a data description registry they will need to know about coverage. For a data analysis library, they might need to know about cost implications, supported formats, and terms of use. So I don't think we'll agree on a single description set that will work for all data services. What we need, I think, is an enumeration of types of data service, a common mechanism for describing services, and a common approach for working out the most useful information to collect for each type of data service.

¶ I had a think about this and came up with a model which I offer to you now (Figure 1).

Regardless of type, I see a four step process going on here:

§1. The first step is getting the list of services of the type we're interested in, be that data repositories, analytic engines or whatever. To support this we might need an ontology of service types, or we might rely on setting up catalogues for each service type, e.g. re3data.

§2. The second is to filter that list so that, hopefully, we are left with the ones that would actually do what we want. So for each service type, we need to anticipate what users' mandatory requirements might be.

§3. If this gives us more than one candidate, and we only need one. there is a third step which is to choose between them. Human requirements can be quite subtle, so often the choice has to come down to interpreting a free text description. Automatic selection of a service has to rely on structured information. But either way, we also need to anticipate what preferences a user might employ to choose between services.

§4. Finally, the user needs to know how to interact with the service. For human-facing services, this is normally the URL of the web form. Tools would normally have selected services by the APIs they have in common, but if coming to this for the first time, the tool author might have to look up the API and implement it before continuing; then the tool, again, needs to know the URL of the API endpoint. There may be authentication layers to get through as well.

In summary then, I think we need a list of service types, and for each service type, the properties that users would most commonly use as requirements and preferences, and lastly the additional information they need to use the service. Let me illustrate this from some initiatives I've come across.

¶ The main reason I'm here is to represent the Metadata Standards Catalog Working Group. You might be aware that the Metadata Standards Directory is the evolution of the DCC Disciplinary Metadata Catalogue. When we designed that, we worked through a similar list of things to consider, based on who would be using it and what they would be looking for.

We were starting with a much more contained problem than today's topic, since we were primarily interested in just one list: metadata standards.

§ Within that, what would be the key filters? Well, the group we targeted initially was research support staff, such as subject or data librarians. When researchers came to them, the easiest and most useful question for them to ask would be 'What is your subject area or discipline?'

§ As the catalogue was aimed at human beings, the next thing to consider was the description. What sort of information would the users need in order to decide whether the standard was relevant for documenting their data? Well, primarily they would be interested in the type of data the metadata standard was intended to document. We could have gone down the route of an exhaustive typography of scientific data types and tagging standards accordingly, but for one thing that would have been probably an endless amount of work, and for another, probably much more detail than the aforementioned librarians would need. Instead we went for a high level description of

the sorts of data targeted by the standard. We also put in notes about who maintained the standard, recent version history, and so on, to give an impression of how well maintained the standard is, that being an important tie-breaking consideration.

§ Lastly, we thought very hard about what users might need to know about a standard in order to use it. This included:

- documentation: user guides or specifications
- associated vocabularies
- crosswalks
- tools to help people work with the standard
- services with expertise in using it

So these are all included in our data model and exposed in the records we display about metadata standards.

Now we're looking at redesigning the catalogue for a broader group of users, including automated tools, so we're reassessing what we'll need in order to achieve that.

¶ But rather than speculate on how that might turn out, there is another example I want to draw on. There has been a push in the UK, mainly from the Equipment.data people in Southampton, for universities to make certain information about them automatically discoverable using what have been termed Organisational Profile Documents or OPDs.

Autodiscovery works either by

```
http://www.bath.ac.uk/
```

```
<link rel="openorg" href="http://www.bath.ac.uk/organisation-profile.opd" />
```

or a redirect from `http://www.bath.ac.uk/.well-known/openorg`

Conventions for specifying datasets: `http://opd.data.ac.uk/docs/datasets`

- Each institution lists its data resources *either as downloads or endpoints, along with RDF description*
- Correct 'output' identified by subject and format/API (`conformsTo`)

The OPDs themselves are simply collections of RDF triples; the recommendation is to publish them in Turtle format. ¶ Here's the one for my institution, the University of Bath.

http://www.bath.ac.uk/organisation-profile.opd

```
@prefix oo: <http://xmlns.com/foaf/0.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
# And so on...

"profile.ttl" rdf:type oo:OrganizationProfileDocument ;
    foaf:primaryTopic <http://www.bath.ac.uk/#org> .

<http://www.bath.ac.uk/#org> rdf:type org:FormalOrganization ;
    rdfs:label "University of Bath (The)" ;
    foaf:homepage <http://www.bath.ac.uk/> ;
    # And so on...

<http://www.bath.ac.uk/equipment/equipment-list.xls> rdf:type dcat:Download ;
    oo:organization <http://www.bath.ac.uk/#org> ;
    dcterms:subject <http://purl.org/openorg/theme/equipment> ;
    dcterms:conformsTo <http://equipment.data.ac.uk/uniquip> ;
    dcterms:license <http://creativecommons.org/publicdomain/zero/1.0/> ;
    oo:contact <mailto:research-equipment-sharing@bath.ac.uk> ;
    oo:corrections <mailto:research-equipment-sharing@bath.ac.uk> .
```

Now, the point about this is that the Equipment.data team can search for OPDs, and when they come across a resource with the right `subject` and `conformsTo` properties, they can harvest the data and add it to their catalogue with a minimum of human intervention.

I suspect the reason it has gained the traction it has is because it has struck a sweet spot between standardization and flexibility. The method of expression is generic, and there is a minimal set of key values, like the link relation and the subject, to trigger the desired behaviour.

I worked for a while on the UK Research Data Discovery Service, and had some brief chats with Adrian Cox about whether we might be able to use OPDs. I don't think our prototype was up to it at the time but actually it wouldn't have been so hard to do. Given an `openorg` theme of, say, `research-data`, we could have looked for a OAI-PMH or CSW endpoint, or, say, a downloadable XML dump of the catalogue, and set up basic harvesting accounts registered to the contact email address. And that would have been quite nifty.

¶ To finish off with, I'll go through the remaining questions we were sent.

Is the *OWL–S framework* appropriate for describing data services?

- Promising for fully automated interactions with services *but is this the problem that this group wants to solve?*
- Would rely on service providers writing the profiles – *not scalable for a discovery service to write them all. It is more the sort of thing the service could link to if available.*

- Could we get by with something simpler? *Seems designed for a world without registries/catalogues; if we factor registries in, somewhat simplifies the problem.*

How should we *describe inputs/outputs*? Do we need to?

- Implicit in the nature of the service
- Implicit in support for standard protocols
- Controlled list of 'known' protocols, otherwise link to formal definitions (WSDL, OWL-S, …)? *I think, again, it is important not to try to re-invent the wheel, but to work with technologies that are already in use.*

Are they *stateless* or *state-based* services?

- Client state: depends on the complexity of the transaction
- Server state: reproducibility versus machine learning?

What about *scientific workflows*?

- Not sure about using them to describe services/processes
- But data services might be used as part of a scientific workflow
- Talk to vendors about incorporating data service registries into products?

Do we need *discipline-specific classification* of data services?

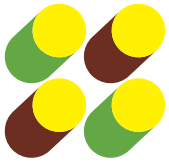
- *The broad categories of data services probably do not need to be so specific, but within each category we will need a more specific classification.* Need category-specific classifications to which disciplines contribute – *within each category, one list with unique classes, but the disciplines need to tell us what classes they would find useful. E.g. in lists of data repositories, tags of data types held; in lists of analytic libraries, the sorts of transformations performed.*

Is *scalability* a functional characteristic?

- Would be useful to know about
  - maximum input size
  - order-of-magnitude turnaround time if not 'instant'

Is *citation* instrumental in making services discoverable?

- Citation more appropriate for workflows (myExperiment, SageCite)
- Certain expectations for cited material also helpful for services: persistent URLs and global IDs, version control…



The Metadata Standards Catalog Working Group is part of the Research Data Alliance, which is supported by the European Commission, the US Government and the Australian Government.

For more information, please visit https://rd-alliance.org/groups/metadata-standards-catalog-working-group.html