New Project: Metadata for data collections

[This is a re-post. I put this up a couple of days ago but my hosting provider lost the server for a day and had to restore from backup.]

At the Australian Digital Futures Institute we've landed a contract with ANDS (the Australian National Data Service) to write some software specifications for key bits of infrastructure for the data commons. I'm going to be doing most of the initial work on this one, and I'll use my blog and Twitter to communicate what I'm up to as I go, as per the communication plan for the project. I know that some of my regular readers are interested in this stuff.; please comment if you have anything to add.

This first post will look at the scope of the consultancy, and talk about some of the major issues that I think we'll need to resolve.

Scope & Deliverables

Our contract is to p**rovide specifications for software applications to help institutions manage metadata about data collections and project plans to build them**. There are two models for these applications. The first is a stand-alone system, while the second is using the existing institutional repository to store metadata – here's what the project plan says:

> The proposed [agreed now] process is for ADFI staff to prepare at least one scoped and costed software specification, with an agreed development methodology for a:
>
> > […] standalone metadata store that can be used to augment either an existing institutional repository or data store, and which will manage metadata about data objects and data collections. This metadata will need to complement the existing object-level metadata. The interface should be designed to be web-based and easy to use by repository managers without specific technical skills. The metadata store should be able to generate collection descriptions as RIF-CS (http://www.globalregistries.org/rifcs.html) and make these available for harvesting using OAI-PMH and/or direct harvest of XML.
> >
> > Source: ANDS-internal document.
>
> Additionally we will develop one or more specifications for providing metadata capture solutions based on existing institutional repository software.

So the main thing we'll be looking at is a kind of repository or registry-like thing where institutions can describe their data collections, and publish those descriptions to Research Data Australia. If you visit that site you can see the data model that's being used by ANDS with four classes of thing. Here's a snapshot of how it looks today:

> Collections (877)
>
> Where a collection is a useful grouping of physical or digital items.
>
> Parties (260)
>
> Where a party is a person or organisation that has some relationship to a collection, service, activity, or party.

Services (1)

Where a service is a mechanism for gaining some kind of access to or information about a collection (or items within a collection).

Activities (2)

Where an activity is an undertaking or process related to the creation, update, or maintenance of a collection. [I think projects are activities - PS]

This model, which is expressed in a schema known as RIF-CS, is based on a yet-to-be-approved ISO Standard (ISO2146). RIF-CS implements quite a different approach from the way most Institutional Repositories (IRs) are structured, where the repository items themselves are the only primary objects.

In an IR you typically have metadata which **refers to parties and services and so on using text-strings**. Parties, services and activities **are not things** in the repository (speaking in general here, there are some exceptions, and increasing people have some status as objects in repositories like Fez). A publisher, which is a party, will be described in an XML document conforming to some metadata schema (in Australia this is typically Dublin Core, MODS or MARC XML) using its name expressed as a string. If two records refer to the same party using different strings then things start to get messy. And as we all know, there is one kind of party around which strings proliferate and cause confusion: people. There is an ANDS effort under way to provide services for describing people outside of a repository so you can refer to them using some kind of ID, but that service is a way off still.

RIF-CS is described here as an interchange format (it is, after all the The Registry Interchange Format - Collections and Services) The which I gather was originally cooked up to support the Global Registries Initiative. But I think some of the people are thinking of using it as a storage format within repositories. If we decide do that we will have to do so carefully. For example, you would not want to store this example RIF-CS as a single repository object, trust me, as the file contains a number of things that really should be treated as discrete objects in a repository, including metadata about people and projects.

Issues for a new standalone metadata store

If the goal for the standalone metadata store is to support the abstract model behind RIF-CS then one of the challenges of this project will be working out how to support this kind of a model. How do make sure that parties and so on are described once, and as accurately as possible? And where that fails, make sure there is infrastructure to assert that two differently described parties are the same, and two identically described parties are in fact different. If you want to refer to a party that is not described yet how will we support that? Will people and objects and so on be primary objects?

In discussions so far, we have tentative agreement between our ANDS stakeholders and ADFI and on one key point; as far as possible we'd like the stand-alone metadata store to be Linked (Open) Data ready. This means that all Collections, Parties, Services and Activities would have URIs, and the metadata store would allow data entry that uses the URIs behind the scenes. (I checked with Scott Yeadon at ANDS and he tells me that RIF-CS 'keys' can be URIs and are expected to be globally unique, even though this is not entirely clear from the schema documentation). This is a user interface challenge, but if we can pull it off we should be able to avoid stuff like this, where a mistyped string results in two parties where there should be one:

http://services.ands.org.au/home/orca/rda/list.php?group=&class=Party&page=2

- Australian Institute od Marine Science
- Australian Institute of Marine Science

So this is all pointing to a repository which uses RDF to describe things, drawing on appropriate vocabularies for relations and values, such as FOAF for parties, the bibliogrpahic ontology for documents, and maybe the Dublin Core Collections for data collections . We could, of course build a classic database-backed system with a database schema that reflect the abstract model directly, but that's very inflexible and not easily extensible.

(We don't have a lot of experience with large-scale RDF at ADFI, but I am thinking that in addition to the RDF, and a triple store to keep it in you would probably still have a high performance index of the repository using Apache Solr to provide the search/browser interface.)

Issues with adapting an IR

If we work on storing data collection metadata in IRs then there will be additional challenges. All institutional repositories in Australia already support Dublin Core metadata, at least in their OAI-PMH feeds. But a lot of repository software in use in Australia is limited to storing metadata as plain strings. Because of this, most of the repository people are used to thinking in terms of flat metadata models serialised in XML, not networks of relations, RDF style. Anything we do to augment IR software will have to fit in with the way IRs are used now, mainly for document content. There are lots of design challenges there for the information model and user interface. We have started thinking about this in detail for a particular repository platform at a particular uni, more on that soon.