ANDS Metadata store: starting point

This is the second post about the ANDS-funded metadata store work we're doing at ADFI. The project now has a Trac site where we will be tracking progress and keeping notes; the site will be open to the public to read, to make it as easy as possible to reach a wide range of stakeholders, although there will be a few documents we have to keep under tighter control. The Trac site will be mainly used as a project wiki – but I will put in some job tickets and use the milestones feature to track what happens between our (mostly) weekly project meetings. This week's milestone is up now.

In this post I'll reveal some detail about our starting point (though not the name of the institution(s) we'll be working with) and follow up on some of the feedback I got from ANDS staffers to my previous post. Scott Yeadon raised quite a few points, some of which I will get to in future posts and/or project plans and wiki pages.

Starting point

ADFI staff met with ANDS stakeholders late in 2009, and we have agreed that a good starting point will be to focus on one of the 'additional' deliverables first. The core deliverable is a project plan to build a stand-alone metadata store, with an option to write extra plans for add-ons or customisations to existing repository software such as DSpace or Eprints should any organisations want to keep metadata about data collections in their IR. It happens that there has been a fair bit of work at least one institution (University X) in Australia where they do plan to keep metadata about collections (and maybe parties, and so on) in their IR; they are running the VITAL software which was associated with the ARROW project. So that's our starting point: a project plan to write some open source software and supply configuration files and customisation, and documentation for an ARROW repository so that University X and other sites running the ARROW suite of software can participate in the data commons, and submit data to research data Australia via their IR.

The VITAL software is a web interface to Fedora, with configuration to index a Fedora repository and display. You can see it in action at the home of ARROW, Monash University. Some sites use bit of free software called VALET to put things in to the repository.

VALET has a simple design which I like – it allows you to design a form or forms to capture as much metadata as you like and configure a set of really simple approval steps. When a user starts adding metadata about a new object, the system saves the in-progress data by the simple expedient of serialising the form data to XML. Moving to the next step of a workflow just requires the application to put the saved data back into the form. When a user with the correct rights adds the item to the repository, pre-configured XSLT stylesheets run automatically to transform the serialised form data to whatever is required, usually MARCXML and Dublin Core.

The ARROW project sponsored a replacement for VALET called Squire which I reported on in 2008, but so far nobody has used it in anger. For this project I think it might be good to use Squire, or a something like it rather than VALET; we're discussing the pros and cons with ANDS and University X.

Over the next week I will be putting together a skeletal draft of a project plan based around the proposed architecture ready for stakeholders at ANDS and the IR and eResearch communities to comment now that people are back from their summer holidays.

Issues

Some of the things which will need to be resolved are:

1. **Which OAI-PMH provider to use?**

   Metadata will get from the VITAL repository to Research Data Australia via an OAI-PMH feed, but there are a few open source toolkits to choose from. We will need to support at least one, maybe more. As ANDS staffer Xiaobin Shen reminded me in the comments of my last post, one consideration will be whether or not the provider supports deletions, not all do. This will require careful testing before we commit to one provider or another.

2. **What data model will be used?**

   At the moment, all the VITAL repositories in Australia that I know about have a very simple data model. Each item in the repository has a 'master' metadata record usually in MARCXML, sometimes MODS (they're effectively the same) with derived metadata in Dublin Core. There may also be datastreams, usually PDF files. At this stage there is no formal content model, so the datastreams could be called anything and it's up to humans to make sense of them; there's no guaranteed way to tell whether a PDF is an abstract, or the whole record, a preprint or a published version of a paper, for example.

   It's a bit hard to get information about VITAL unless you're a customer; the product page is currently sporting a copyright statement from 2008 and the brochure (PDF) is big on tropical fish and short on specifications so what I report here may need correcting.

   The version of VITAL which I think most sites are running in Australia is 3.x. It uses Fedora 2 which doesn't have formal content models. Fedora 3 has a formal mechanism for describing content models which means that you could describe the parts of an object, and their role in the object. From what I can gather VITAL 4 which I saw demonstrated in late 2008, and was released in March 2009 has content models too, but they are more about how to display an object than describing the relations between its parts. Perhaps someone could elaborate or correct this in the comments?

   My working assumption is that for this development, the idea will be to stick with the way VITAL 3.x works, without worrying about content modelling which is fine here because this is not about complex objects with lots of data, it's about metadata about data collections where the collections themselves will usually reside elsewhere, which brings us to the sub-question.

   What goes in the repository and what is stored elsewhere?

   There's a real chicken and egg problem here. I gather that eventually the NLA will be running a party-identifier service based on People Australia, so when that's established we won't be typing names into metadata forms any more, we'll be linking to an ID. So in the abstract model behind RIF-CS the party management just goes elsewhere. I wonder if the same could happen with activities (every project has some kind of web site now, so why not point to an RDF endpoint hosted on the project website or just to the project web site as an identifier) and services (not sure what to do about these, but then I'm not really sure yet what services are in this context).

   Question is, if we want to get a system running now, what's the best way to identify parties in a future-proof way? I discussed a related issue in a blog post for CAIRSS about NicNames and People Australia. Maybe NicNames can play a role here in the short term?

   One of the design patterns I mentioned in that post, using an index to associate names in a repository with an identity service like NicNames via an index is expanded in a paper I wrote for the New Review of Information Networking. At the moment I can only link to this version which is not open as the publisher has not responded to my questions about the OA policy so I have yet to deposit a version in Eprints.

3. **Which metadata format to use?**

   Scott Yeadon made it clear in the comments to my last post that RIF-CS was designed as an interchange

format only and that it is not yet stable, which sounds like good reasons not to use it as a storage format. but I have confirmed reports that others in ANDS are thinking otherwise, and are encouraging IR managers to put RIF-CS in the repository; I'd like to hear their side of the story too. Stability aside, if RIF-CS has what it takes to describe a collection of data then it might be an OK storage format.

I'm not aware of all the alternatives but one that I have heard mentioned by and ANDS person is the Dublin Core Collections Application Profile. What else is out there in use for describing data collections and are there other data-collections registries harvesting from those descriptions in the rest of the world?

So this is an open issue for now; I hope we can get some consensus on a good data model for storing metadata about data collections (and the other entities).

4. **What configuration is needed in VITAL? I think we need the following:**

   - Display for items (are we going to have parties and services and actions as items as well?)
   - Index configuration for VITAL's Solr Index.

   And If the data collection resides in the repository should there be a collection object **and** a collection-description object or just one object with both collection and description?

5. **What kinds of APIs do we need?**

   VALET can be used to integrate with other systems via XML files which are deposited in a directory and picked-up by the ingest workflow of VALET, so they can be curated by data librarians, a technique which I think was developed by Simon McMillan at UNE in the RUBRIC days. We can certainly implement this, but should we have a web (or other) API for a system such as a grants database to add a new item as well? Should it be AtomPub or a simple post, or SWORD or something else?

This post was written in OpenOffice.org, using templates and tools provided by the Integrated Content Environment project and published to WordPress using The Fascinator.