I've been working with software designed for institutional repsoitories (IRs) now for quite a while. At RUBRIC we helped our network of partners to evaluate repository software. Follow that link to read about DSpace, Eprints, Fez and VITAL.

Having worked with four and a bit applications (the 'bit' is the Mura application which is not complete), and with the RUBRIC project coming to an end in December 2007, I've started to think about things that could be improved in future generat ions of repositories.

One area where I don't think any of the applications are doing the best they could is in helping repository managers in enforcing best practice for keeping stable URLs for objects, and giving things **useful identifiers for use in different contexts** (No, It's not enough just to have handles).

A lot of people say that best practice for identifiers is that they should be semantically null. That is, there's no meaning in them that humans think they can decode. This really makes sense for published resources. If you base an identifier on anything that might change or you might get wrong then you have a problem; once an identifier has been published then someone may have used it and it is not good to invalidate the old identifi er. You either have to have infrastructure to maintain old identifiers or be very strict about not renaming things. For example, if part of the name is the name of your company or university what happens if it gets gobbled up by a bigger company or university or your depart ment moves between institutions?

(See this easy to read discussion of the issue.)

On the other hand, having identifier with some semantics can be really helpful in some cases, and it's essential for things that people need to manage by hand, such as word processing files.

One example where identifiers with semantics are nice is blogging, where it feels right, and is really useful to have the classic blog /year/month/day/ as part of the path. But there are issues with the last part. I sometimes create identifiers for blog posts that are a bit funny looking by mistyping.

So if I had been blogging when I was thirteen, I might have made an identifier like:
http://ptsefton.com/blog/1978/05/16/mispellled_id

Note the identifier is not just he misspelled bit at the end, **it's the entire URI**. Norm Walsh spells this out. As he says:

> As names, “HB88”, “http://norman.walsh.name/knows/what#nikon-5700”, and “newscheme:x:y:n5700” are entirely equivalent. They're just strings.
>
> http://norman.walsh.name/2006/07/25/namesAndAddr esses

For me this is an OK identifier – I figure I can keep the domain registered and make sure that as I migrate the blog across software platforms I can keep supporting those identifiers. There's little chance of error as the year / month / day part of the identifi er is generated for me by a machine. And I really don't see what harm the date in the path is doing. (And if I decide to move things around I can always use handles to help me)

My policy at my web site is that once I have published an item I won't go back and change the identifier, because it's out there and one of my four regular readers may have bookmarked it or tagged it, or posted about it. (I will fix typos within a post if I see them,

but apart from that all updates will be noted at the top of the post).

But **I often make mistakes** in generating identifiers. So I'm going to remove the error -prone bit and change the way I make ID's starting with this post.

The last part of the Identifier on posts is usually generated from the file name I use to write the document.. The filename on this one is `repository-interfaces.odt` which is not at all what it ended up being about. See, that's an example of a ***meaningful identifier gone wrong***.

So I'm going to replace this code which generates the last part of the ID from the filename:

```
thisPostName = os.path.splitext( os.path.split(self.path)[1])[0]
```

With this, which uses the time:

```
thisPostName = str(datetime.datetime.now().time())
```

The time will reflect the point at which I decide the post is nearly ready not the time it goes onto my server. For my purposes this is **unique and it's machine generated** so there will be no spelling errors (there's only one of me to hit the button so I can't do two posts at the same millisecond).

At the moment I make a new document in ICE by clicking “Create new Writer document” , and choosing a file name.

As I refine the process I use for blogging, it may be that I can let go of the idea of having an identifier I need to manage at all. I could let ICE create a new document with a GUID (Globally Unique ID) and tell it the title. That way, if the document evolves and the title changes then I will see the new title via ICE, and I will see the item in any of the categories I've added it to using our forthcoming tagging feature – I never need to see the filename, and it may as well be an opaque number.

Some other examples:

For photos

I'm happy to have opaque names for individual files because Picassa does a good hob of showing me my photos.

For music files

I use ITunes and virtually never look at the files – happy to have it handle the naming when it imports a CD, and it's almost irrelevant to me that songs happen to have semantically useful filenames because the metadata does all the work and I never see the files.

For item in an institutional repository

These can (and usually are) be represented by a semanticall y opaque identifier such as this one. You'll have to go and look at it to see what it is :-)

http://eprints.usq.edu.au/archive/00002653/

But the actual paper for item 2653 uses an identifier with a bit more semantics:

http://eprints.usq.edu.au/archive/00002653/01/Sefton_etd_2007.pdf

This is not ideal, as naming the paper is done by humans. I have worked through some

the issues to do with the naming of papers in USQ's repositories with the humans who run it. The software doesn't help them as much as it should. I think the datastreams (eg a PDF) should have opaque identifiers at the server side, but if a user downloads one the repository software should assign it a file name that helps the reader to manage it. So the filename for a paper like this could be: usq_2653_sefton_2006- 1. Or something. At least this gives someone a clue what's in the paper.

For PDFs downloaded from online sources

I used to leave them lying around my Desktop, so if repositories assigned useful names to downloaded files (not their online identifiers) that would really help.

But I bet I'll soon be using a tool like Zotero to pull documents out of web sites and manage them for me, so I need never see a file name.

So here's a draft principle.

Identifiers should have semantics only under some circumstances:

1. For items managed by computers identifiers should be semantically null except where semantics are absolutely inarguable and guaranteed to be correct. (Eg a time-stamp for a blog entry)
2. For items managed by users, or which might need to be managed by users later, on their own computers identifiers should have useful semantics, but we should work hard on building tool-chains that mean that user's *don't have to manage identifiers at all,* anyway.