

Repositories....

...and the known unknowns

Setting the scene

“Reference models, why bother?”
- Bill Olivier

Key tasks

- Construct typology / ecology of repositories.
- Emerging from that typology identify common repository services and distinguish these from domain specific services.
- Identify what kind of services these repositories will offer and consume.

Current practice

- Many existing software platforms for repositories...
- ...with widespread deployment.
- Not developing software & systems from scratch.
- eFramework needs to relate to current practice.
- Reference models must accommodate existing systems.

Repositories in other activities

- Repository issues arise in other strands e.g.
 - Item banks in assessment
 - ePortfolios as repository
 - document management for course validation
 - ...
- Repository reference models will overlap with other reference models.

Levels of abstraction

- Reference models may exist at different levels.
- May be high level communicative tools e.g. ontologies.
- Or low level specifications e.g. SCORM.
- Acknowledged this but no further discussion.

Difficult issues

- End users are joining up networked and desktop services to suit their own requirements.
- Personalisation is becoming a reality.
- Services may be activated at multiple points in a workflow.
- How does this relate to repository typology / ecology? Not clear.

Representation

- Use of UML for gathering requirements usecases is questionable.
- Not trying to build monolithic software applications.
- Aim is not to develop software but identify services.
- If reference models are communication tools other forms of representation more appropriate e.g. mind maps.
- Some way away from these issues.

Some consensus...

- DLF approach appears to be as useful as any other.
- Inclusive definition of “repository”.
- A little dissent...
 - Is a database a repository?
 - Managed and trusted part of Rachel’s definition.
- Struggling to see functional difference between national LO repository, item bank & community image store.
- Is such inclusive definition of “repository” useful?

A comparison of repository types

- A national LO repository, e.g. JORUM.
- An assessment item bank.
- A community image store, e.g. Flickr.
- All use similar abstract services...
- But the way these services are instantiated varies enormously...
- As do the rules and policies associated with these repositories.

Rules and policies

- How do rules and policies relate to reference models?
- In the way they influence the instantiation of abstract services.
- Each rule or policy e.g. student access rights, must have one or more abstract service associated with it, e.g. authentication, authorisation.

Reference Models — Why Bother

- Communication tool
 - between domains
 - new developers, repository implementers
- Evolve to reflect practice, not necessarily to drive it.
- Gap analysis

The danger...

- Don't need to retrofit a reference model to what we already know.
- Is it constructive to focus purely on the abstract?
- Focusing too much on reference models may distract us from real problems that need to be solved.
- We may just reinvent the OAIS model.

The unknown...

As we know,
There are known knowns.
There are things we know we know.
We also know
There are known unknowns.
That is to say
We know there are some things
We do not know.
But there are also unknown unknowns,
The ones we don't know
We don't know.

- The Rumsfeld approach to reference models

For example...

- We don't know what kind of API we need to deposit into repositories.
- Flickr and Fedora have published APIs that anyone can write to.
- Can not do the same thing for Dspace or ePrints for example.

The solution?

- Use OAIS as our high level repository reference model.
- Use this as a communication tool across domains.
- And to help identify problem areas - the known unknowns.

The known unknowns

For example:

- Deposit API
- handling complex objects
- packaging
- federation
- identifiers
- integration with other systems

Deposit - a known unknown

- Known specification relevant to deposit service binding
 - WebDAV,
 - OKI OSIDs,
 - JSR 170 & 283,
 - SRW Update,
 - Flickr API, Fedora API, ECL,...

Way forward

- Support posts will arrange a meeting of a small number of developers (e.g. ePrints.org, Dspace, Fedora, Greenstone, Intrallect) to agree a trial deposit API.
- CETIS and DRP support to arrange second meeting looking at whether OAIS is appropriate reference model for JISC community.

Unknown unknowns

- ?