How to Implement an OAI publishing registry in 1 day

Patrick Dowler
Canadian Astronomy Data Centre

The back story...

- CADC didn’t get much into the registry game early on
- used a registry implementation from AstroGrid project
  - massive/complex java war application
  - embedded XML database did not fit operational model (redundancy/HA/LB)
- then a sysadmin re-org of a filesystems on a bespoke internal server trashed the XML database…

- … and our OAI publishing registry was dead and non-recoverable

- … then searchable registries started doing their semi-annual full harvest and could not verify any of our resources still existed

Implementing OAI publishing in a day

- recover the content
  - Markus helped a lot by writing a script that pulled all the current records from the MAST registry
- minimise the scope
  - only do what is needed by other IVOA registries
  - one authority
  - modest number of services
  - future content updates only made by me
  - BUT: implement good test code (~re-usable)
- shortcomings (identified from OAI spec and rofr.ivoa.net validator):
  - recovered XML records are in ivo_vor format: cannot support oai_dc output format #fail
  - OAI timestamp parsing is odd, my code is more lenient #fail

Implementing OAI publishing in a day

- OAI verbs: 6 different requests to support

- with the limited scope, 4 of them output static documents:
  - verb=Identify
  - verb=ListMetadataFormats
  - verb=ListSets
  - verb=GetRecord&identifier=<ivo_id> (1 doc per service)

- the recovered content are all the OAI GetRecord documents
  - fix docs with my new OAI registry URL
  - Identify doc and our registry doc have same <ri:Resource>
- implementation:
  - name each doc to match verb or identifier
  - read file and rewrite with correct OAI doc envelope
  - updating a record: have to change timestamp in two places

Implementing OAI publishing in a day

- two of the OAI requests have dynamic content:
  - verb=ListIdentifiers
  - verb=ListRecords
  - filtering by date (from & until)

- implementation of ListIdentifiers:
  - read every xml file and and extract the OAI <header>
  - maybe filter by date(s)
  - rewrite with correct OAI doc envelope and sequence of <header>
- implementation of GetRecords:
  - read every xml file and extract OAI <header> and <metadata>
  - maybe filter by date(s)
  - rewrite with correct OAI doc envelope and sequence of

<record><header>...</header><metadata>...</metadata></record>

TODO

- define a back end storage API
- move code to https://github.com/opencadc

- work on or accept contributed back-end implementations that are less dumb

- would really like a back end that was also RegTAP