Max Planck digital library

Code Discussion

eSciDoc Syndication Service

Aims and Expectations

eSciDoc Syndication service should

- ensure a common and extensible way for generation and access to the eSciDoc last updates via wide range of the most up-to-date web syndication feed formats.
- deliver a well-formed and standardized feeds to be correctly parsed both by web-browsers and software aggregators
- be implemented as stand-alone web service
- be applied for web syndication purposes in all eSciDoc solutions (PubMan, Faces, VIRR).
- Use other eSciDoc services for data retrieval and transformation

Technologies and components

- Apache Digester (parsing of the feed definitions)
- EsciDoc Search&Export service (retrieval of the PubItem lists for the feed generation)
- JIBX transformation (item-list to ROME objects)
- Project ROME (feeds generation)
- JSP (presentation level)
- Organizational Unit Handler (retrieval of the Organizational Unit List used by presentation level)

R5 features

- Feed classes:
  - recent releases in repository
  - recent releases for a specific Organization Unit
- Presentation level
- <link ... rel="alternate" type="application/rss+xml"/> in html headers
- Feeds are always actual (no TTL for feed, no caching)
- Feed entry always has PubItem XML in content element (where the feed type specification allows it)

Implementation details

- Generic syndication feed type of the ROME is used to ensure the mapping to the all supported feed formats
  - pro: easy implementation
  - contra: less metadata granularity granularity
- Data retrieval, only in PubMan contex:
  - HTTP request to the Search&Export web service. CQL query directly defined in the feed definition XML
- Transformation: XML=>PubItemVO=>SyndEntry
- Caching:
  - HTTP response headers (cache-control, Expires) for the browser cache handling is implemented, but not used due to requirements for the start-up R5 release
  - an internal caching implementation were planned, but will be replaced with the PubMan common solution for web-proxy caching (i.e. squid server)

TODO

- PubMan GUI for feed support. JIRA ticket AS-739
- Common way for the definition and implementation of the feed classes like:
  - recent releases for a specific publication (complex, with many stages retrieval from repository)
- Feed classes specifications for other eSciDoc solutions (Faces, VIRR)
- Caching issues
- Separate mapping for RSS/Atom feed types for more metadata completeness (if needed)
- Move syndication entires transformations to the transformation service

Links

- Syndication Manager web presentation
- PubMan Web Syndication Feeds on Colab
- Subversion
  https://zim02.gwdg.de/repos/common/trunk/common_services/syndication_presentation
- Path to the Design in EA: /Desing Model/Use Case Realization/SyndicationManager
- Wikipedia: RSS/Atom
- ROME project