Invenio as a platform to implement the SCOAP^3 Repository

Open Repositories 2014, June 9-13, Helsinki

Samuele Kaplun and Wojciech Ziolek

Outline

- Introduction to SCOAP^3
- The use case and requirements
- Getting data
  - Proprietary Invenio APIs (*batchuploader)*
  - OAI-PMH (OAI Harvest)
  - FTP servers (BibTasklet)
- Automatic metadata normalization
- UI/UX
- Search Engine Optimization

Introduction to SCOAP^3

"SCOAP^3 is a one-of-its-kind partnership of thousands of libraries and key funding agencies and research centers in two dozen countries. Working with leading publishers, SCOAP^3 ***is converting key journals in the field of High-Energy Physics to Open Access at no cost for authors.****"*

(from the http://scoap3.org)

Scope of the Repository

“SCOAP3 Articles ***shall be available open access*** *without limitation in time, and their widest re-use shall be possible. They shall be accessible without any barrier on the publisher’s website and shall be delivered in a timely manner (as defined in Section 3.2.2)* ***to a repository operated by SCOAP^3****, for further distribution and re-use under the applicable License(s) as per Section 3.1 (e.g.,* ***redistribution to institutional repositories of participating institutions or subject-specific repositories****).”*

(from the **TECHNICAL SPECIFICATION**)

The use case and requirements

- 10 publisher feeds to aggregate:
  - metadata
  - PDF and PDF/A
  - XML representation of papers
- 3 months to realize it
- Administrative tool to evaluate publishers compliance with contracts
- Aggregation tool to disseminate to 3000 participating libraries and beyond

Ingestion of data into the repository

- 2 publisher pushing MARCXML via *robotupload API*
- 1 publisher providing MARCXML via OAI-PMH
- 3 publishers deposit *.zip packages* into FTP servers to which we have been granted access

Robotupload API

- Publishers push via HTTP POST request MARCXML compliant to our profile
- 1 publisher exploit new callback support, for deposit confirmation
- to set this up:
  - **CFG_BATCHUPLOADER_WEB_ROBOT_AGENTS**
  - **CFG_BATCHUPLOADER_WEB_ROBOT_RIGHTS**

Robotupload API

```
CFG_BATCHUPLOADER_WEB_ROBOT_AGENTS = invenio_webupload|Invenio-.*|MuleESB
CFG_BATCHUPLOADER_WEB_ROBOT_RIGHTS = {
    '89.202.245.160/27': ['IOP', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    #'62.50.9.128/28': ['IOP', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '62.50.0.0/19': ['IOP', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '149.156.74.9': ['Acta', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '149.156.74.52': ['Acta', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '149.156.74.80': ['Acta', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '149.156.74.87': ['Acta', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '149.156.74.100': ['Acta', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '149.156.74.115': ['Acta', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '149.156.74.138': ['Acta', 'SCOAP3', 'SCOAP3 Repository', 'DELETED'],
    '137.138.0.0/16': ['TEST'], # useful for testing
    }
```

Robotupload API

- New IP-based authorization by supporting network ranges (i.e. CIDR)
- Pro tip: note the addition of **DELETED** to allow publisher to delete their records.

OAI-PMH

- Publisher providing MARCXML but not directly matching what *bibupload* understands (due to usage of *marc:* XML namespace)
- We handled this in Python via a bibfilter

OAI-PMH

- See: https://github.com/kaplun/scoap3/blob/master/hindawi_bibfilter.py
- Pro tips:
  - `from invenio.bibupload import find_records_from_extoaiid`

    i.e. use the same algorithm bibupload uses to know if an harvested record already exist in the system or not.
  - `from invenio.bibrecord import record_add_field, record_xml_output`

    i.e. generates output MARCXML using bibrecord library.

FFT

- Note: both publishers pushing data to us via *robotupload* or making it available via *OAI-PMH* agreed to provide proprietary Invenio FFT tags, to let the repository automatically *pulling* the corresponding **PDF** files.

FTP Server

- Ad hoc library to:
  - connect to FTP server
  - discover new .zip packages and retrieve them
  - unpack and discover .xml representation of papers, alongside PDFs
  - building MARCXML metadata from .xml representation
  - upload MARCXML with .xml and PDF

JATS

- Nice story:
  - some publishers are moving towards standard XML representation of their papers, i.e. JATS http://jats.nlm.nih.gov/
- This simplified our implementation of a Pythonic converter XML -> MARCXML

Automatization

- FTP server crawling automatized via BibTasklet
    - https://github.com/Dziolas/scoap3/blob/master/bst_elsevier.py

  i.e. micro bibtasks that wrap simple functions and execute them regularly

Automatic metadata normalization

- Thanks to new BibCheck module:

```
[check_crossref_timestamp]
check = crossref_timestamp

[check_iop_arxive]
check = iop_arxive_fix

[check_iop_issn]
check = iop_issn

[check_arxiv_prefix]
check = arxiv_prefix

[check_add_publisher]
check = chk_add_publisher
```

Automatic metadata normalization

- E.g. to correct systematic typo

```
def check_records(records):
    """
    Amend the records to rename 037__9 arxive into 037__9 arXiv
    """
    for record in records:
        for position, value in record.iterfield('037__9'):
            if value in ('arxive', 'arxiv'):
                record.amend_field(position, 'arXiv')
```

Automatic metadata normalization

- Perfect for:
  - correcting systematic errors
  - translating metadata pushed from outside
  - completing metadata with external sources
- Checks are automatically applied to new and modified records
- http://invenio-software.org/wiki/Development/Modules/BibCheck

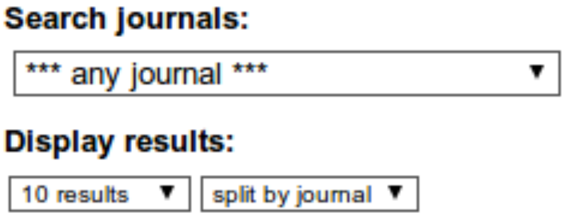
UI/UX

- The repo is an administrative tool
- User oriented functionalities reduced to the minimum
- Everything not needed is disabled (when possible via WebAccess)
- Corresponding URLs for disabled functionalities lead to 404

- Customized:
  - **webstyle_templates.py** (thanks to WebStyle)
  - **websearch_templates.py** (thanks to WebStyle)
  - **webinterface_layout.py** (through a hack)

UI/UX

- Some small improvements:
  - Renaming of “collection” to “journals” by overriding **websearch_templates.py**
  - Javascript hack to not clutter URL when, from home page, user start searching without selecting any collection
  - Publishers are giving us XML with MathML. So we enabled MathML in MathJAX



Search journals: *** any journal ***

Display results: 10 results, split by journal

Search Engine Optimization

- Identified site with Bing and Google:

```
<meta name="google-site-verification" content="
mLqufkdPNxUHXFW4obCfN5NJXr4sD_SlnvsOla7RZAE" />
<meta name="msvalidate.01" content="
EA9805F0F62E4FF22B98853713964B28" />
```

- Enabled BibExport Google SiteMap generation

```
[export_job]

export_method = sitemap

collection1 = SCOAP3 Repository

fulltext_status =
```

Search Engine Optimization

- Enabled OpenGraph and Scholar export in HTML

  HEAD:

```
<!-- GoogleScholar -->
<meta content="Sphere-level Ramond-Ramond couplings in Ramond-Neveu-Schwarz
formalism" name="citation_title" />
<meta content="Bakhtiarizadeh, Hamid R." name="citation_author" />
<meta content="Garousi, Mohammad R." name="citation_author" />
<meta content="Elsevier" name="citation_publisher" />
<meta content="10.1016/j.nuclphysb.2014.05.002" name="citation_doi" />
<meta content="Nuclear Physics B" name="citation_journal_title" />
<meta content="884" name="citation_volume" />
<meta content="408-437" name="citation_firstpage" />
<meta content="2014" name="citation_publication_date" />
<meta name="citation_online_date" content="2014/05/12">
<meta content="10.1016/j.nuclphysb.2014.05.002" name="citation_doi" />
<meta name="citation_pdf_url" content="http://repo.scoap3.
org/record/2395/files/main.pdf" />
<!-- OpenGraph -->
<meta content="Sphere-level Ramond-Ramond couplings in Ramond-Neveu-Schwarz
formalism" property="og:title" />
<meta content="website" property="og:type" />
<meta content="website" property="og:type" />
```

Conclusions

- New features introduced to Invenio
  - Network range protection for robotupload
  - MathJax-based support for MathML
- Strongly exercised (and consequently debugged and improved)
  - **BibCheck** to automate metadata normalization and enrichment
  - **BibExport** for SEO
  - **BibTasklet** to automate ad-hoc data inputting
- Publisher-specific code to fetch, crawl, parse packages now available as a shared project with fellows at INSPIRE: https://github.com/inspirehep/harvesting-kit