Symplectic Repository Interoperability

John Fearn
Co-founder/CTO
john@symplectic.co.uk
High level goals

- Make full-text openly accessible through automation and high user engagement
- Monitor and report on OA compliance and engagement
- Empower librarians to administer in their way
Philosophies

- Use open standards for open ecosystems
- Repository platform agnostic
- Work out of the box
- Simple to use
- Fit into existing library workflows
Long history of experience

**RT1:** first bidirectional CRIS/Repository integration at Imperial College (2009): DSpace repository deposits skyrocketed. Improved and repeated across ~40 top universities. EPrints and Fedora 3 supported.

Based on RT1, the **SWORD 2** standard was developed and published by a Symplectic team member.

**RT2:** Second generation connectors released 2016, ~80 universities now on **RT1/RT2**. DSpace, EPrints, Figshare and Hyrax now all supported.
Technical challenges, and their solutions

- A smooth end-user experience
- Platform-agnostic solution
- Flexible 2-way data mappings
- Work out of the box
- Cater to conflicting stakeholders
- Support multiple connected repositories
- Report in one place
- Don’t overwhelm repository system
Technical challenge: a smooth end-user experience

Solved by implementing:

- real-time automatic 2-way metadata synchronisation
- A uniform experience for all supported repository platforms
Technical challenge: platform-agnostic solution

Solved by abstracting repository capabilities into common interaction building blocks and using open standards

- Create, retrieve, update, list all, list modified, etc...
- Using SWORD 2, OAI-PMH and platform-specific extras
Technical challenge: flexible 2-way mappings

Deposit using either system. Records will automatically appear in both systems.

Use Elements to detect duplicates in your repository.
Technical challenge: flexible 2-way mappings

Solved by

- Inventing a powerful yet simple “xwalk” mapping language: no XSLT
- Exposing all Symplectic data faithfully
- Providing a self-service mapping editor and integrated test environment, with version control

Harvest crosswalks

<table>
<thead>
<tr>
<th>Repository</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Title</td>
</tr>
<tr>
<td>Provenance</td>
<td>Volume</td>
</tr>
<tr>
<td>Volume, Issue</td>
<td>Issue</td>
</tr>
<tr>
<td>DD.MM.YYYY</td>
<td>YY-MM-DD</td>
</tr>
</tbody>
</table>

Deposit crosswalks

<table>
<thead>
<tr>
<th>Repository</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Title</td>
</tr>
<tr>
<td>Author</td>
<td>Author</td>
</tr>
<tr>
<td>ORCID</td>
<td>ORCID</td>
</tr>
<tr>
<td>Linked grant fields</td>
<td>Linked grant fields</td>
</tr>
<tr>
<td>Subsequent deposit</td>
<td>Subsequent deposit</td>
</tr>
<tr>
<td>Filed added by</td>
<td>Filed added by</td>
</tr>
</tbody>
</table>
Technical challenge: flexible 2-way mappings

- Revert to any previous working version
- No custom plugins, code or configuration required in the repository
- Test and troubleshoot in seconds
- Fully supports all XML and JSON metadata standards, including CERIF, METS/MODS, and any proprietary/custom metadata extensions
- Persistent identifiers and funding metadata included
Technical challenge: work out of the box

Solved by:

- providing rich working default mappings for all supported systems
- working closely with repository platform development teams to influence the development of their APIs

Typical time to get up and running with a fully-functioning RT2 solution is ½ hour. Customers usually spend longer enabling their APIs.
Technical challenge: cater to conflicting stakeholders

Solved by:

- Native support for multiple data sources: Librarians are free to edit repository data without fear of affecting data curated by other types of Symplectic user.

- Librarians specify which repository fields Symplectic may and may not update, and under what circumstances.

- Verification and curation can be performed either side, and by independent stakeholders, without conflict.
Technical challenge: support multiple institutional repositories

Solved using native support for multiple data sources

- Multiple repositories of the same type
- And of different types

Seamless central reporting and oversight of Open Access and Open Data engagement across specialist or departmental repositories.

E.g.

- Figshare used as Open Data repository
- EPrints used as Open Access repository
Technical challenge: central reporting

- Every connected repository’s metadata faithfully pulled into a central and directly queryable reporting warehouse
- In-application custom reporting based on repository data
- Specialised ‘OA Monitor’ dashboards and reports, with support for multiple configurable OA policies
- Enable researcher prompts to deposit ‘In policy’ publications
- Identify ‘OA champions’
- Manage article-level exceptions and opt-outs
Repositories only care about a subset of the metadata managed by a CRIS system. Most repositories cannot cope with the high rate of CRIS data changes seen at many of the larger universities.

Solution: downstream system specific “relevance” settings that automatically filter out uninteresting changes (and publications).

E.g. reduced load on UNSW Fedora 3 by > x100
Thank you

come and say hi