FOLIO/ERM

FOLIO Days Göttingen - Background and Progress Update
Ian Ibbotson, 2018-04-267

https://creativecommons.org/licenses/by/4.0/
Audience Interests and Goals

- **General FOLIO?**
  - Should we focus on the platform, its aims and affordances?
  - Or on the specifics of ERM?

- **ERM Context?**
  - Do we need to talk about the problems faced by eResource librarians?
  - Any specific interests or concerns?
Ceci n'est pas une pipe.
How many grapes went into the wine....
Some terminology

- **KB / Knowledge base** - generic computing term with library specific meaning. Here we are taking this to mean the “Publications Knowledge Base” - a shared understanding or model of what (Title, Coverage) is being published, by whom, on what platform and how it is packaged up for sale by vendors. The focus of the KB is “Packages and Titles (With Coverage)”

- **ERM** - we are taking this to mean the information about what MY institution thinks it has bought or has access to (Titles and Coverage), from whom, on what platform, for how long and under what conditions. The focus of the ERM is Agreements which relate of a list of titles with some access and coverage, on a platform, to a license. Agreements may be subscriptions, but could represent other things like Hathi Trust collections.
Terminology (2)
Why a [FOLIO based] ERM (Causes)

- The problems faced by eResource librarians extend well beyond keeping a list of what packages have been bought and what they contain.
- (KB View: Centering the problem on the vendor) Mismatch between the title/package tracking provided by vendor KBs, and the tracking / audit / cyclical tasks undertaken by libraries as customers. This mismatch causes much use of spreadsheets as a mechanism to cross system boundaries.
- [Honest] There is concern about the extent to which vendor managed knowledge bases reflect what was bought vs what the current provision is.
- [Generous] There is a perceived lack of transparency about what was bought and what is provided, and vendor managed KBs do not always add clarity.
- Some issues caused by vendor induced fragmentation in ERM space (KB Lock-In)
Some questions kept coming up:

- Can you list all the titles we buy multiple different ways? -- A concern that we were being sold the same thing many times in different ways
- Can you help with our renewals process
  - Generating our big spreadsheets
  - List me all the agreements coming to a close in the next 2 months
- Can we link usage data to provide informed decisions (Spreadsheet linking) -

Why KB+ / LAS:eR / KB+ 7 (Approaches)

- Introduce the “Subscription Agreement” and “Issue Entitlement” (Agreement Line Item) concepts into the domain model. Essentially modelling the institutional “Side” of the ERM problem and a compliment/reciprocal to Package/Title.
- Separate out coverage from IE, allowing PCA and Subscription coverage discriminators, and PCA to be described in package/agreement
- Allow connection/attachment of a “License” to the Subscription.
- We can now link from title details to license properties.
- Allow analysis and interpretation of license terms, and link to titles.
- Subscriptions allow renewals workflows to be modelled and tracked
Approaches (Experimental)

- Introducing variables into package or agreement descriptions: `${SubEndDate}`
- Use of tag-like mechanism to track “Core” status or “Class” of item
- Package/Agreement level coverage statements to express PCA rules for different classes within the package
- Track moving core status for PCA purposes.
  - Programmatically generate coverage based on explicit statements and PCA policy
Why FOLIO/ERM - What’s wrong with what we have?

- Each of the existing solutions live as a Silo, and ultimately have the same problem as their predecessor systems - use of spreadsheets and other transient files as a lingua franca to move between KB, Finance, Link Resolver, Discovery and other systems.
- KB+, LAS:eR, GOKb all have “just enough” requirements to justify being a service in their own right, but all need to have standard functions and deep connections to other systems.
- We have tried to address this problem with interoperability and standards, and that has been successful to some extent
- But it has not led to a more integrated or smooth user experience, and we still move files around -a lot-.
How does FOLIO help address this?

- FOLIO provides an integrated but loosely coupled environment with shared infrastructure like authentication, authorization, reporting, usage stats, ...
- So we don’t have to build “Foundational” infrastructure like authentication for each app, and can focus on the actual problem the app is intended to address.
- Which means we can focus on each app “Doing one job well”, but can integrate and coordinate at a system level to make (For example) an ERM talk to KB, Finance, Discovery, Link Resolver, Notes, Workflow, etc.
Approach Taken...

- Originally, we were going to build the ERM solution outside FOLIO, but with the intention of integrating eventually.
- This approach would reduce the dependency upon the FOLIO environment that all the tools and widgets needed would be available within the project timescale.
- But this approach also introduces the risk that we might develop in a direction not compatible with FOLIO.
- So the revised approach was to accept the risk introduced by the dependency, but work collaboratively with FOLIO from the outset.
- It was felt that close working relationships would mitigate any risk introduced by FOLIO being “In Development”
Work to date

- Workshops - Including the kick-off meeting.
- Initial domain modelling and implementation
- Gathering test data, defining file formats
- Prototyping in FOLIO environment, understanding where conflicts are likely to occur.
- Backlog Development
Domain Model Walkthrough
Major evolutions from existing models

- Agreements not Subscriptions
- Agreement Line Items (Not IssueEntitlements) can be Packages, Package items, or explicit “PlatformInstances”
  - An agreement can just name a package and automatically accept the vendor view of the world (Essentially vendor KB solution)
  - An agreement can closely track the titles in a package, but maintain “My” definitive list
  - An agreement can consist of entirely bespoke titles which have no link to a package.
- Coverage Statements are separate to title list entries (unlike KBART)
  - to allow for multiple coverage statements, coverage gaps, and to remove the assumption that resources will always have coverage - ebooks and other material types. Allow description of PCA rights at Package/Agreement level
Data....
Major Data Sources

- Package Data / KB Data
  - Raw KBART files?
    - A staple of the community for a long while
    - Necessary but not sufficient for the kind of ERM we would like to build
  - Processed by intermediary data curators / Consortia
    - Like -- KB+ / GOKb / ..... 
    - Rich “Header” information - package curator, dates, licenses, consortia, provider, availability
    - Codified PCA and other parameters controlling access
    - Easily structured multiple coverage statements
    - Better support for describing alternate item types
    - Clearer semantics around sibling instance identifiers (print_identifier, electronic_identifier)

Agreement Data

- For FOLIO integration tests
- To bootstrap / migrate users
- To allow us to explore renewals workflows before we explore new agreements.
- More complex than packages
  - Package, Package Item and Platform-Title agreement line items
  - Licenses
  - Other properties

<table>
<thead>
<tr>
<th>Section</th>
<th>AgreementLicense</th>
<th>SectionHeader</th>
<th>LicenseTitle</th>
<th>VendorReference</th>
<th>PackageTracking</th>
<th>AuthorityReference</th>
<th>TrackingDate</th>
<th>SectionHeader</th>
<th>AgreementJournalHoldings</th>
<th>Title</th>
<th>instance.issn</th>
<th>FromPackage</th>
<th>Medium</th>
<th>Media</th>
</tr>
</thead>
</table>
Towards iteration 1 - Landing Page?
Example of great current practice::
Jisc Collections KB+ Dashboard
Mismatches with FOLIO we are exploring...

- 4 levels of nesting, panel layouts, navigation
  - Seen in other areas such as invoice line items, innovative solutions and extensions
- More complex search form and facet lists
- “Dashboard” layouts
- Separation into Apps - granularity
  - Similar issues seen with eHoldings, amplified here
  - ERM (Agreements, Agreement Line Items, Packages, )
  - Licenses
- Subscription Agreement domain entity - different to eHoldings
ERM Dashboard

Agreements
You have 12 current agreements. Workflows you can launch from here:
- Create a new empty agreement and add content later
- Search packages and create a draft, trial or live agreement

Upcoming Renewals
No agreements ending within 8 weeks
Click here to update settings.

Recently Edited Agreements

Subscribed Content
Your current agreements provide access to 36455 individual electronic resources.
Quick Search: _______________________

Packages
You are currently selecting titles from 23 packages

Content Issues
There are currently no reported issues accessing electronic content. The world is happy.

Licenses
Your Licenses
Recently Edited Licenses
<table>
<thead>
<tr>
<th>Name</th>
<th>Agreement Type</th>
<th>Start Date</th>
<th>End Date</th>
<th>Renewal Date</th>
<th>Next Review Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>A new agreement (TRIAL)</td>
<td>TRIAL</td>
<td>01/01/2018</td>
<td>31/01/2018</td>
<td>01/01/2019</td>
<td>01/01/2018</td>
</tr>
<tr>
<td>A new agreement (DRAFT)</td>
<td>DRAFT</td>
<td>01/01/2018</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary
The decision to work in FOLIO from the start

- We have to deliver an ERM - adding a new platform into the problem space is adding a variable: Our concern about platform maturity has not been an issue at all
- Environment is modern and easy for developers
  - Some disclaimers for the ERM team here - We’re off reservation in some of our tooling (like eHolding and finance to some extent tho, and this is a positive)
- Documentation is readily available and easy to follow
- Managing test instances can be slightly challenging
- The fully running system requires some resource (But what do you expect for a functional library services platform)
Velocity

- Our current velocity GOOD
- Platform is not causing friction
- Implemented full domain model - with requirements to date
- Implemented functional package import (KB+)
- Implemented “Ziffer” application
- Implemented example “Agreements” list and search
- Implemented example dash
- Implemented CODEX interface to provide subscribed content search
Upcoming

● Development Path (Steve leading)
  ○ Backlog agreed
  ○ Start work on backlog issues
  ○ Explore “Subscribed Content” search
    ▪ Agreement, Package, Title, Coverage, Core/PCA in one line
  ○ More work on deployment (folio.k-int.com)
  ○ Expand work to development team

● Exploratory Path (Ian leading)
  ○ Explore Workflow issues in more detail
  ○ Explore Finance link issues in more detail
  ○ Explore eHoldings link in more detail
Overall

- Platform has supported development with very minimal friction. Compared to RICE velocity is good. Loose coupling major factor.
- Platform does have some idiomatic requirements that developers can find tight.
- Design constraints much bigger factor than technical constraints
- Very likely that decision to work in FOLIO from the outset was correct if eventual goal is full participation in FOLIO ecosystem of apps.
- Community engagement extremely good.
- High confidence of delivery.
Followup Channels

ERM Subgroup Wiki

https://wiki.folio.org/display/RM/ERM+Sub+Group

RM Discuss

https://discuss.folio.org/c/sigs/rm
Questions
Thankyou

ian Ibbotson
ian.ibbotson@k-int.com