

DspaceUserGroupBergen2006

A page for notes and thoughts from DSUG Bergen 2006 - <http://dsug2006.uib.no/>

If we all use the tag 'dspacebergen2006' for our photos on flickr, and links on del.icio.us (and so on, please add others), we'll have a convenient source of [photos](#) and [web resources](#) related to this user group meeting.

Contents

- 1 [Presentations](#)
- 2 [Collection Structuring B.O.F](#)
- 3 [Submission and Workflow B.O.F](#)
 - 3.1 [\(1\) Need for more information to enter and store for several fields](#)
 - 3.2 [\(2\) Need for specification of input-forms.xml on document-type rather than collection or community.](#)
 - 3.3 [\(3\) Possibility to move items to other collections](#)
 - 3.4 [\(4\) Possibility to upload items directly into multiple collections](#)
 - 3.5 [\(5\) Store a URL to a bitstream rather than the bitstream itself](#)
 - 3.6 [\(6\) Propose example values and/or default values in input-forms.xml](#)
 - 3.7 [\(7\) Possibility for hiding bitstreams and relating bitstreams to each other.](#)
 - 3.7.1 [example \(1\)](#)
 - 3.7.2 [example \(2\)](#)
 - 3.8 [Problems with input-forms.xml](#)
- 4 [AddOn and Patch writing B.O.F](#)
 - 4.1 [Reasons for customisations / extensions / addons](#)
 - 4.2 [Current customisation code management practices](#)
 - 4.3 [The Addon mechanism](#)
 - 4.4 [Other tools and addon architectures](#)
 - 4.5 [Road map for component/extension management](#)
- 5 [Installation, Deployment and Configuration B.O.F](#)
- 6 [Network APIs B.O.F.](#)
 - 6.1 [Standards vs direct exposure](#)
 - 6.2 [Packaging](#)
 - 6.3 [AuthN AuthZ](#)
 - 6.4 [Transactions, Versioning](#)
 - 6.5 [Purpose](#)
 - 6.6 [Finally](#)
- 7 [Community Discussion Session](#)
 - 7.1 [1](#)
 - 7.2 [2](#)

Presentations

Presentations will be available on the conference website shortly.

Collection Structuring B.O.F

What logical structures are available to us?

- Content type is entered in the keywords
- No particular structure is imposed (Cambridge)
- Each community can structure on their own needs
- By research areas
- By file or document type
- By time
- By the way the university is structured; departments – though some may be closed so those documents become legacied and/ or placed in other departments.
- Decentralized, by faculty structure of the university (Cambridge – most likely decision)
- Originally organized by units within the university (Edinburgh), but lists are large without many entries; did not reflect the whole university. Interdisciplinary groups did not fit into the traditional academic structure; set up separate communities for them. Can choose to submit single items into several collections.
- Authors can select one or more collections for their submissions (MINHO); can appear in the collection of the library as well as the community. Advise authors to limit levels of hierarchy.
- Does this matter with Search capability? perhaps this is a theoretical discussion but not really critical
- How to structure the e-thesis for harvest? Separate collection for theses and student reports.

How deep can our hierarchies be?

Naming conventions – best practice

Multilingual problems and solutions

- Collection names
- English vs. native language of authors/ institutions
- Researchers want to reach the world, so they use English names; would be preferable to list in multiple languages
- Institutes and Faculties must be named in one language though text may be in two languages; preference for English for international adoption

Administrative policies

- At least one mandatory step: Library metadata review and validation (MINHO)
- Reserved? Open? Peer reviewed material only? Community determined? (Is this worrisome? Will some things that should be preserved be lost?)
- Faculty don't want to deal with any of the administrative elements.
- What about one collection, rather than divided into smaller collections?
- Bergen: browse on titles, areas and dates

– Amy Hale

Submission and Workflow B.O.F

(1) Need for more information to enter and store for several fields

Many bibliographic systems, like Aleph, have resolutions for this and call it subfields.

Example (1) is with citations where multiple fields are needed. This means that input-forms.xml should be enhanced to support 'subfields' for about any possible character field possible.

Example (2) is the author-field where there is a need to store also email address and some unique number (Digital Author Identifier). This unique Author ID is needed to make lists of works from the same author easily, to generate lists of author works made in multiple universities and eg to identify that an author has published with different names (before and after marriage p.e) but in fact is the same person. Furthermore the option is required to call external routines to show a list of authors (like from SAP or from a nation-wide author-thesaurus as available in the Netherlands at Pica). The University of Leuven (Lieven Droogmans) demonstrated a possible solution for what is required on authors.

(2) Need for specification of input-forms.xml on document-type rather than collection or community.

Although the current collection or community level is very nice and a big improvement, many institutions store multiple document types in a single collection. And each document type requires a different set of metadata fields to maintain. Now institutions create workarounds to organize collections by document-type. Specification at document-type level therefore would be needed.

(3) Possibility to move items to other collections

Possibility to move an item to another collection during the submission process. This to correct mistakes during submission. Now that is impossible and the author has to re-enter all data

Possibility to move an item to another collection when finalized in dspace. This is needed and handsome when eg. re-organizing collections.

(4) Possibility to upload items directly into multiple collections

Would be nice during submission to directly be able to specify multiple collections. Currently this must be done afterwards, after submission, by administrative people. One of the examples was also 'automapper' (not in the standard dspace), but manually during submission are needed as well.

(5) Store a URL to a bitstream rather than the bitstream itself

Is handsome when the bitstreams are huge (video streams) or when the bitstreams are available elsewhere in another repository or at a commercial publisher.

(6) Propose example values and/or default values in input-forms.xml

Examples given during data-entry greatly clarify how users should enter fields during submission.

Proposing defaults would greatly reduce the amount of work for a user during submission while leaving the option open to deviate from the default.

(7) Possibility for hiding bitstreams and relating bitstreams to each other.

example (1)

Uploading a word document which must be hidden in the UI and uploading (or generating) the related pdf-file which must be shown.

example (2)

Replacing a bitstream in dspace because of eg. format upgrade, like a better version of pdf, or because the author delivers an improved version of the bitstream. The newer bitstream must replace the existing bitstream but the previous bitstream should remain available as previous version in the archive.

Problems with input-forms.xml

Not possible to enter eg Norwegian strings in input-forms.xml. Presumably this should be an installation-problem or setup problem but nobody knew at the moment. Suggest to ask it as a question /problem in the dspace-tech mailinglist ?!

– Peter Ruijgrok

AddOn and Patch writing B.O.F

Reasons for customisations / extensions / addons

- interop to particular local services
- small presentational changes (usability or branding related, look and feel with other local sites)
- more systematic presentation-layer changes
- functionality that is outside scope of the core DSpace

One common problem is managing JSP changes - particularly tricky. Probably will NOT be in the scope of an addon mech (not in the sense of merging different JSPs anyway!). Perhaps Manakin (XSL/Cocoon presentation layer framework) will help with this (??)

Current customisation code management practices

- Use of patches - but head can change in the meantime so requires maintenance
- Tapir initially used a different code tree but now developed using patches
- In some projects for minor changes a record is kept and the changes reapplied to later versions of the head

The Addon mechanism

- to ease the installation of additional components
- ideally would be some kind of runtime plugin, but due to the current state of affairs, the idea is simply to be able to manage the addition of components at build time (as the DSpace installation process is build-based in any case).
- details are in Richard Jones' presentation, but the general idea is to put addon code into an addon component template, and then run various ant targets to install this together with the core DSpace code
- we are getting to a workable basic soln for an addon mechanism, but this needs refinement
- a working group on this problem has been suggested - needs involvement from more than just Richard J, to work on
 - addon mech itself (XSLT skills for config merging esp desirable)
 - reference implementation
- the wider developer community can help by refactoring existing features to use the addon mech and also the plugin manager when appropriate
- work is also needed on making the addon mech and the plugin manager work together well
- work on the DSpace config system generally may help with that

Other tools and addon architectures

- OSGi (Open Services Gateway Initiative) - swiss army knife of componentised Java applications. Standardised way to define Java functionality and add/swap implementations at runtime. Bundles expose interfaces/abstractions and require others. Big but complex. [OSGi link](#) [another OSGi link](#)
- Maven - better build system than ant for straightforward/std projects, and great for dependency management of e.g. jar files providing underlying services/utilities. But would it be adaptable enough for, or help with, a complex/custom build like the addon mech?
- Also - not discussed but may be relevant -
 - Java Plugin Framework - <http://jpf.sourceforge.net/>
 - Java services - <http://java.sun.com/j2se/1.5.0/docs/guide/jar/jar.html#Service%20Provider>

Road map for component/extension management

- Short term
 - move to a workable system for component management (i.e. addon mechanism)
 - move to coding and building more modularly
 - move to a compartmentalized DSpace
 - use a minimal DSpace core ?
- Long term
 - move to a more standard and/or powerful extension management system

– Liam Lynch

Installation, Deployment and Configuration B.O.F

The group consisted of some who have installed DSpace dozens of times and have niggles with it, new users who have installed it once and had some problems, and users who had never installed it.

Here are some ideas that came out of the discussion:

- It might be good if the installation documents (as well as the application interface) was translated into different languages
- A pre-install check tool (like ./configure perhaps) included to check for pre-requisites
- A post-install checker to ensure the application is healthy
- Possibly OS specific packages (e.g. an RH RPM)
- A specific install FAQ on the wiki
- Instructions on what to do if the FAQ does not answer the question (e.g. email dspace-tech)
- Support for other DBs (e.g. mysql)
- A tool to help edit dspace.cfg and input-forms.xml (maybe in the DSpace admin interface)
- Simplification of the stylesheet to allow for easier customisation
- A document with a 'tour of DSpace' for new users experimenting with it
- A tool to check the DC registry against the input-forms so that internal server errors are not thrown when you update input-forms but forget the DB registry
- install_configs is confusing - which config files do you edit (dspace-src|dspace-src/config or dspace/config or the templates?) and when do you run install_configs

– Stuart Lewis

Network APIs B.O.F.

The group consisted of some who have implemented web services around DSpace, some who are going to soon, and many more who were interested to learn about the issues and opportunities.

Standards vs direct exposure

The issue here is to consider the pros and cons between web services that conform to open standards (WebDAV, SRU, OAI) and those that expose the DSpace logic directly (e.g. Generate WSDL directly from DSpace application classes). There are a number of differentiators: -

***Interoperability** The standards based approach is obviously far stronger, based on an object model that is shared, rather than DSpace's own.

***Comprehension** Whilst standards are generally understood by more people, they may require more effort in learning to achieve a particular individual task. This is one of the reasons people have implemented WS services in the past.

***Stability** Because standards based approaches are abstracted from DSpace's implementation, the web service APIs supported should remain more stable across changes to DSpace. Exposing code directly means the API changes with the code.

***Speed of implementation** Writing support for a standard such as WebDAV is a major undertaking, whilst with tools such as Apache Axis, client and server stubs can be generated far more easily.

What is the minimum set of standard that would be needed to provide ws access to the DSpace core?

This is not particularly a question of SOAP vs REST - the LNI is a standards base, high level service that supports both.

Packaging

Packaging is seen as a key aspect of web service access to DSpace, and the package manager plugin that allows support for arbitrary packaging standards was welcomed as necessary.

AuthN AuthZ

Most current implementations have web services consumed by other server clients, and use prior knowledge of the consumer to establish trust (usually with certificates). No-one in the group has heard of a better solution.

Cross system authorization requires a shared model of rights and groups. LDAP is most usually used to store this information, and organizational hierarchy used as the model for groups (department, research group) and rights levels (PhD, Professor etc). Conversion is required to translate between DSpace permissions and this institutional model.

There was an interest in whether Shibboleth could provide a good solution for AuthZ.

Transactions, Versioning

Many envisaged use cases would require long term transactions or locking (e.g. external agents to perform metadata analysis / file migration).

The automation of content modification highlights the requirement for simple linear versioning.

Purpose

The current uses of web services with DSpace is usually moving packages in or out, e.g. for federation. Future use cases envisaged included: -

- Web interface as a web service consumer
- Add-ons (e.g. preservation plugins) as web service clients in order to simplify management of installed code, and to spread processing power requirements better.
- Customized collection portals with non-standard indices and interfaces.

Finally

Members of the group are to add a description of their efforts on the NetworkInterfaces page on this wiki

– JimDowning, 2006-04-21

Community Discussion Session

After Peter Morgan and Julie Walkers' Federation and Community update on Friday 21 April, a 30 minute community discussion session was held, facilitated by Jim+Downing. The following are different people's notes on the discussion. If you have a notes from that session, please add them below.

1

- Work by Technical working group is going to be critical. Would like more information on how this group will function and their purpose. Response: No decision made yet; will not be the committer process.
- Key role for working group is to establish where the platform should be heading, how that will happen. Steering committee will report on this.
- What's the scope for members to give feedback to the steering committee? How can users interact with them?
- How should "we" make decisions for the community?
- Response: The steering committee will not present things it as a fait accomplis. Steering committee will solicit opinions on the wiki. Or respond to questions personally.
- Do members want a voting process?
- Can there be a questionnaire to identify the needs of the DSpace community? (innovation, governance, etc.)
- What do the institutions want to commit to?
- Is there any thought to paid memberships? Subscription fees? Resources? Participation? Are there other kinds of subscription contributions? E.g. Developer time?
- Consensus is that their institutions would be willing to make a commitment to the DSpace community at some level.
- Possible Fees: 500-1000 euros per year (under a thousand euros). Personal or institutional memberships would allow for wider spread. Any fees need to reflect the affluence level of different countries, perhaps through multiple tiered memberships.
- Formal membership with low financial contribution would provide a reasonable solution; useful for getting new functionality but getting formal position in the decision making process.
- Sentiment that people want to have a "right" to speak.
- There are national users groups that are organizing and considering direction as well. e.g. Dutch users group is currently looking for a central place where all Dspace is harvested; how can they deliver the metadata to the central repository?
- How do users get representatives on the technology working group or the steering committee.
- HP and MIT own the code. Shift to non-profit entity would transfer code to them; universities would need to sign over their code to the non-profit. Ownership to the code might be able to stay with the developer of the code with a license for use from the non-profit.
- How will technology working group be formed? What do the users want? Can they have a say in it?
- Lack of coherent communication between users and technical direction. (handling of metadata); needs to be a stronger dialogue.
- Would like to see a technology roadmap. (needs a commitment for effort on the technology)
- If Roadmap had a tasklist assigned to it; you could find volunteers who might agree to elements of the roadmap.
- Short term and long term road maps, taken from the users – what new features they want. Members vote which add-ons and features they value most.
- Big move towards modularity, but there are many different needs and interests; risk bloat of the core package.
- Committer group vs User driven decisions on modules. Pitfalls
- Will this take away from the users with a management committee running things?

- Innovation will be based on short term problems; long term preservation is an issue to be solved.
- Mission statement should include statement about preservation.

Three tiers:

- High level steering committee
- Technical working group (architecture and roadmap)
- Code community

My thoughts:

Notable absence of all members of the steering committee and the technical working group. Is there a huge gap between these groups and the users? How can groups responsible for "destiny" questions properly reflect on the needs of the organization and its users?

– Amy Hale

2

- How do we provide feedback to the steering committee? This meeting is an opportunity to investigate the concepts
- Would a questionnaire to the DSpace community to define the needs of the participants be a good idea?
 - What sorts of needs? e.g. Innovation
- Federation Membership?
 - Some kind of feedback or return would be necessary
 - Financial membership is a reasonable option. Formal involvement in the decision making procedure would be desirable
 - Financial commitment "not very high" - ~€ 100s not 1000s of Euros
 - Tiered subscription - costs dependent on economic environment
- What do you get out of a subscription based model?
 - Dutch DSpace User Group has a central DSpace harvested, and defines together what they do locally
- Legal issues:
 - Ownership of the code could go to the resulting non-profit organisation
 - Could institutions own their own code, and licence to DSpace?
- The Technical Working Group:
 - Communication between users and developers
 - develop the "Road Map"
 - There will be different local requirements of the working group
 - A subscription model may make it easier to define the Road Map
 - Not yet known how it will be formed - the steering committee must advise
- What about having short term road maps initially. How could features be requested to appear on it?
- It may be appropriate to vote on features and addons to be included before each Feature Freeze for releases (The community reiterated its interest in the AddOn Mechanism)
- The idea of a community wide commitment to preservation was raised

– Richard Jones