CommitterMeeting24June2008

DSpace committers meeting - 24th June 2008 - HP Labs, Bristol

Introduction / agenda

Preliminary Discussions: Tue AM

- Introduction, Agenda
- Planning and Scheduling 1.5.1
- Communications and bug tracking - JIRA?
- Fedora collaboration possibilities: To what extent should we attempt to align data models? Where do we envision possible points of contact?
- OAI-ORE: In or out of scope? Influence on DSpace+2.0 data model?
- Completed work and revisions to 2.0 goals to date.
- Status of UI support: Will 2.0 work be exposed in both interfaces?
- Workflow engine: Do we integrate, and if so, which candidates?
- Review of http://wiki.dspace.org/index.php/Frequently+Asked+For
  (lunch break)

Design Planning: Tue PM

- Code structure / layering / pluggability / service manager. Do we SOA? Do we Spring?
- Group guidance on 2.0 data model: goals, anti-goals.
- Generate worklist and rough priority
  (adjourn for sustenance and contemplation)

Work Planning: Wed AM

- Milestones, interfaces, integration points
- Migration strategies
- Communications plan and initial work assignments
  (lunch break)

TBD: Wed PM

Planning and scheduling 1.5.1

Mark D. has volunteered to do release management of 1.5.1. We already have considerable amount of stuff accomplished in branch. There are bug fixes, SWORD implementation fix, xml UI bug fixes, LDAP working in manakin. Mark would like to get release out the door.

How will we advise people to do upgrades? You will need to maintain your customizations and merge them in somehow.

At a point we need to make decisions on maven build process versioning. Anytime you run Maven build package you will be getting the latest updates in the current set up. Using continuous integration system developed by Bamboo.

Richard Rodgers suggested before releasing 1.5.1 we should poll the community to ensure we have identified all the obvious bugs before the release. Mark D has discovered bugs based on MIT's installation, RSS feeds working with xml. Caching issues with RSS feeds. Mark would recommend these be fixed before rolling out. Continuous integration system is building interim release builds. Mark D. believes we should us the interim release build to do installation tests.

Brad states should put together open items list for 1.5.1, determine what level of testing, and how to make is more obvious to use integration builds. Right now have roughly 4 items to be fixed for 1.5.1. Brad will ask the community for input, Mark will drive 1.5.1 release.

Testing of release-Have 3 organizations here running/experimenting with 1.5.1. Will do installation smoke test to validate.

Need declared issues list of what we will close, and running smoke test to get to resolution and release. Brad will drive this process. Foundation and rest of team will drive the survey process.

The old statistics packages now works with Manakin in the 1.5.1 release. Now enough data generated in the logs to actually generate the stats required.

Brad will post the question to the list to identify any other bugs in 1.5.

Mark will put together an email regarding build integration.

Communications and bug tracking

Michele highlighted the fact that 'the community' in general do not like interacting with SourceForge. Th SF interface is not good, and an account is needed.

Should we change to using JIRA? Would JIRA solve the issue of people not joining in? Most people think JIRA would be better, but would it solve the problem?

Mark Diggory highlighted the fact that it is a social / communications issue, not a technical issue. He wants the foundation to empower people to interact and have a voice. The foundation is working with the community to build up a structure of people across the world to help address the communication issues. The wiki is not great as it is hard to expose what is inside, or to search it.

We have information and code in to may places. How can we address this? Richard Jones suggested making the wiki home page **THE** DSpace home page, with all the links in one place. The audience for the wiki is much small, unstructured, and unmanaged. The wiki home page does need work.

Brad's conclusion:

MediaWiki needs to stay - we can't migrate again.** On that note: http://confluence.atlassian.com/display/CONFEXT/Universal+Wiki+Converter

SF deficiencies need addressing - JIRA may be an option.

- - http://www.atlassian.com/software/jira/
- - Full offerings: http://www.atlassian.com/software/development.jsp

We need to work ourselves to integrate the services seamlessly.

Fedora collaboration possibilities

DSpace and Fedora - Collaboration Exploration Meeting - June 10, 2008

Ideas generated during technical brainstorm session:

- Identify high impact scholarly applications (e.g., Zotero) to integrate with our repositories; initiate meetings to move forward
- Jointly mentor the Google+Summer+of+Code project that will demonstrate DSpace running on top of Fedora
- Define common content models

- Map the DSpace 2 data model to Fedora digital object content models;
- Define a common reference data model for institutional repositories
- Investigate adopting common standards (protocols; interfaces; formats) for deposit of content into our repositories (e.g., SWORD, APP, other)
- Investigate scenarios for integrating a open source workflow engines with our repositories; demonstrate workflows that use both DSpace and Fedora
- Define and implement a common event notification architecture (e.g., JMS provider in open source)
- Define and implement common storage APIs (e.g., JSR170, XAM, Fedora‚Äôs Akubra) for interfacing repositories to underlying storage systems
- Joint implementation of the ORE data model to enable sharing and exchange objects from our repositories

- Work out issues around the semantics for content models using ORE (e.g., journal article, image, journal, book, etc.)
- Develop shared services/modules to enable exchange of objects among repositories using ORE?
- Host a web service for creating personal aggregations/collections using ORE?
- Joint support of JHOVE; investigate possible funding strategies to provide support to JHOVE initiative
- Investigate use cases from both communities to come up with shared output ??
- Investigate shared user interface approaches

- Manakin on top of Fedora and DSpace?
- What is best way for to provide an out-of-box repository application that works with both DSpace and Fedora
- Other ways to enable building of lightweight apps on top of our repositories?
- Move toward common architecture for modularization of core software; evolve to a plug-in approach where modules are sharable by both DSpace and Fedora?
  Thoughts:

- Common/compatible event system? JMS? Atom?
- How to map DSpace objects to Fedora to ORE?

- in the abstract data model.
- Mark hopes the data model work done will have certain component that is abstract and defining how the DSpace data model will exist and can use as a guide when new functionality is added.
- Tool for mapping into standards and other systems.
- Jim D- means that will push a data API for storage, and moved up the agenda.
- Rob suggests DSpace developers join the Fedora mailing list to see how the community works.

  ### OAI-ORE

  What is it? - http://www.openarchives.org/ore/

We'll cover this as we go as appropriate.

Completed works and revisions to 2.0 work

Here.

Status of UI work

JSP UI and Manakin - with 2.0 work, do we want to make any support statements about these? Is it healthy to keep both in order to foster competition? How do we maintain feature compatibility in both, and for how long?Do the two interfaces need to maintain an identical set of features? How will the community feel about this?

Issue will need to address is there is a lot of integration at the business logic layer- built into the user interface. There is not a clear separation of services and features. Therefore will be challenging to allow the 2 interfaces to diverge-as it will affect the way the application works.

Aaron would argue to support as many interfaces as possible to be able to generalize a service layer.

Rob-xml and cocoon chosen in move to make UI more modular. Should be able to pull out business logic from UI, and can facilitate using a workflow framework. Rob comments both JSP and cocoon are outdated now and need to support next generation framework and tools within the application.

Workflow engine

Are there any changes to the recommendations that came out of the architectural review board that need review or change?

Mark D.- believes some of the identifier work scoped in the recommendations is obsolete, based on the versioning work done to date.

Data Model

- Wiki page for existing model: http://www.dspace.org/index.php?option=com_content&task=view&id=149
- PDF for recommended model: http://wiki.dspace.org/static_files/0/0e/DSpace-recs.pdf
- Wiki page for recommended model: http://wiki.dspace.org/index.php/ArchReviewDataModel
- We agree the data model should not require hierarchical identifiers, we should call for hierarchical identifiers within an item. Having opaque identifiers for everything as the only source of relationships might not be the best strategy from the preservation viewpoint. If the data-model could be represented in a hierarchical method, than the identifiers should mirror this relationship and be exposed.
- The introduction of ORE does not change the need/desire for a FRBR type data model.
- There are several manifestations of the FRBR model to be reviewed. A simplistic model is illustrated in the 2.0 recommendations report, however this may not be the most appropriate manifestation to be implemented.

- SWAP is actually an intersection between FRBR and the DCMI Abstract Data Model. So what we have is a representation of the "Entities" of FRBR as Work ~~-> Expression -~~> Manifestation ~~-> Item instead as ScholarlyWork -~~> Expression ~~-> Manifestation -~~> Copy, I've started to have serious concerns with this. Explicitly, both "Copy" and "Item" are abstract concepts and not necessarily representative of the "file parts" of our composite or complex digital object. I am concerned that there is a repeated effort to "shoe-horn" Ontologies like FRBR and SWAP into a fixed, overly simplified and strictly linear hierarchies like DSpace's Collection/Item/Bundle/Bitstream and that every time this happens, its a lossy mapping that results in limiting the users ability to accurately represent the structural nature of their composite digital content. In FRBR, Works and Manifestations (and possibly Expressions) are Containers which may contain themselves, thus the following is possible based on the model (Work ~~-> Work -~~> Expression ~~-> Manifestation -~~> Manifestation --> Item) likewise there are many interrelationship properties (predicates) between siblings and between parent child Entities in the model that go far beyond just Containership, thus introducing a rigid structure reminiscent of FRBR or SWAP here can be quite dangerous and confusing. -Mark Diggory (2008-06-30)I find myself returning to an original viewpoint (originally expressed by Richard Rodgers a couple years ago) that These relationships need to be expressed as Metadata (Statements) attached to the DSpace data model (not dissimilar to that of rels-ext found in Fedora). I take this viewpoint further and suggest that at its heart the DSpace data model needs to be rooted in a generic node-property graph that will initially have a more concrete or explicit SWAP/FRBR Application profile enforced on it. -Mark Diggory (2008-06-30)
- Mark comments work in regard to metadata???mark needs to comment further

- Currently, we know that we would like to see Metadata attached to more than one level in the DSpace data model, it is actually already the case that the some of the classes (Community, Collection) contain a getMetadata() method on them and this method mapped to the table columns in the db. We need to go beyond this with a more consistent implementation. -Mark Diggory (2008-06-30)I perceive both the above issue with capturing SWAP/FRBR relationships in the Data Model and the exposure of metadata capabilities on all the data model objects as one-in-the-same. Both "metadata fields" and "relationships between the data model objects" are the same type of "reference" in our Data Model, in the metadata we have "item > predicate > Literal" mappings, in the inter-relationships we have "dso > predicate > dso" mappings. With this recognition that these are similar referencing mechanisms, an underlying implementation that is based first in expressing these mappings about DSpace Objects, their relationship to other DSpace Objects, and their relationship to external resources provides the starting point for bringing together a more wholistic solution to data-storage and representation in DSpace+2.0. -Mark Diggory (2008-06-30)
- Brad comments on the ability to represent and store aggregations based on ORE, and how the data-model we choose might influence our ability to store and display these aggregations.

- Well, I challenge that the ORE model is really just a re-purposing of an already existing mechanism in its RDF roots used for Referencing resources as the properties of other Resources (resources linked via properties). If we want DSpace to handle ORE or any other model, we should approach the data model at this granular a level (resources linked via properties) such that DSpace is more about describing Resources of different types, linking them together with properties and possibly attaching bitstreams to those resources where appropriate. -Mark Diggory (2008-06-30)
- Is there any mapping out there we would want to support that we know of?

- Mark comments there is a FRBR ontology available (http://vocab.org/frbr/core).

- This is actually what SWAP is for SWAP (Scholarly Works Application Profile) -Mark Diggory (2008-06-30)
- Proposed data model as detailed on the wiki. Need to be able to attach metadata at multiple levels, and describe the relationships between the items/bitstreams etc.

- Aaron comments to possibly look at JCR for versioning the datamodel vs. the approach described in the wiki. If versioning is captured within datamodel, it will force incompatibility with other JCR compatible solutions. In order to support Fedora and other JCR implementations- expose the versioning but let another application do the work.
- Rob comments one of DSpaces strength is the data model is easily understood and this in an important concept to carry forward.
- Aaron comments to try and keep the data model as simple as possible.
- Brad comments for each function/service is there an existing tool/service we can use or do we need to build it into our existing code repository? Want to be able to extend the datamodel vs. building all the functionality within the trunk when possible. However, have to acknowledge the relationships exist with the datamodel.

Review of FAF wiki page -http://wiki.dspace.org/index.php/Frequently+Asked+For

- Statistics
- Versioning
- Distributed Community / Collection Management

- Embargo
- Streaming Media
- Electronic Theses & Dissertations (ETDs)
- Support for hierarchical LDAP servers
- Better Windows O/S support
- Branding (name of service)
- Hit highlighting in search results
- The point of reviewing the list is to take any of these requests into account as we make a plan for the "funded" work in regard to the datamodel.
- Comment from Aaron:Putting these features in before refactoring can make the overall work harder.
- Brad- there is going to be work going on that is not the "funded" work, going on at the same time, so we need to agree on how to deal with potential conflicts as a result.Make sure people associated with these various features/projects listed the team is in close contact with.
- Mark- comments that some of the items on this list may not be applicable for the team to consider, given the nature of the request and possible resolution to the issue.

Discussion before Michele left

Q: Jim D - How can we move towards getting proper funding for core development projects?

A: Try to get sponsorship for features, try to get developers to do the development. Soon the foundation will be approaching the community to contribute to a fund (initially to fund the foundation). The foundation wants to provide training and professional services in order to generate revenue. Brad added that this needs to be a transparent process.

Workflow

- Brad outlined the challenges:Right now it workflow is integrated and hardcoded into the core code, but if we want to work with others (e.g. Fedora) then we would want to take it out as a service. How do we want to position ourselves in this area? Do we want to keep going in our current way, pull it all out into some workflow engine, or migrate slowly.
- Rob wanted to clarify the definition of a workflow. For examples is it a normal deposit workflow, or do we include things like the workflow for creating a collection, or some of the offline jobs. What do we mean by "a workflow"?
- Brad considered it as how to string together the steps required for an ingest step (the activities that need to occur, with rules and constraints), rather than going down to the level of moving from page to page in a particular activity. So somewhere in the range of what Rob described.
- Rob defined this as an interaction requiring more than one user, not just small atomic changes.
- Mark explained some past experiences with struts and workflows. Workflows are always very independent from user interaction with a system, so struts would never be a good workflow engine, you should use a workflow engine instead, and interact with that using something like struts.
- There is work in this area in Kuali which is probably worth following up on. (http://www.kuali.org/)(http://confluence.arizona.edu/confluence/display/KUALI/What+is+Kuali+Enterprise+Workflow)
- There is some work in this area with Sakai and assignments. They might be a good source for asking for advice? (http://bugs.sakaiproject.org/confluence/display/FRAME/Workflow+Considerations)
- Brad: Is this an area which is urgent? Does it need to feed into core 2.0 work, or can is be dealt with separately?
- Aaron: Maybe we should address this when we talk about services?
- Rob: The API needs to be for the data model. Lets learn from the situation we have at present - Rob can talk about ths if required.
- Mark: Does the atMire GSOC student have anything to feed in here?

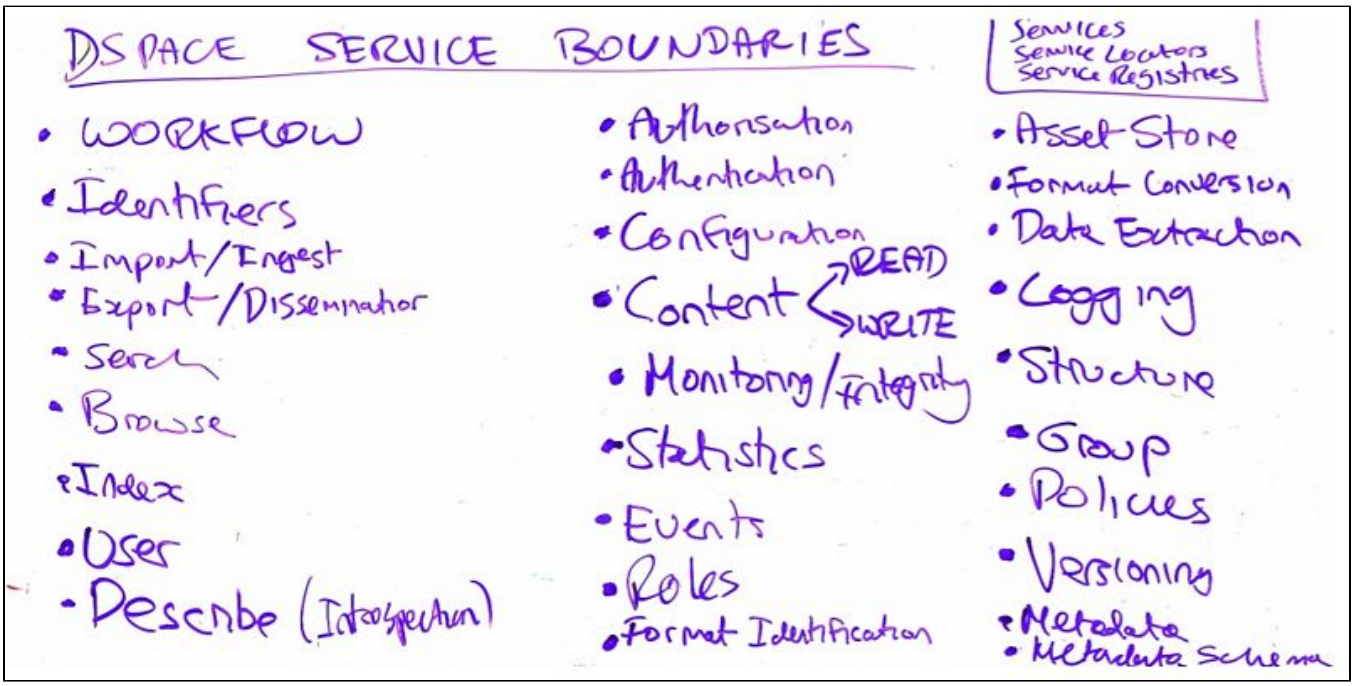
Services

Two activities: 1) List the services DSpace provides, 2) The mechanics, potential toolkits etc.DSpace services:

- Workflow
- Identifiers
- Import / ingest
- Export / disseminate
- Indexing
- Search
- Browse
- AuthN
- AuthZ
- Group management
- Roles
- Policies
- Configuration
- Content

- Read
- Write
- Asset store
- Transformation
- Data extraction
- Monitoring / integrity
- Logging / statistics
- Events
- Users
- Describe (introspection)
- Structure
- Versioning
- Metadata store
- Metadata schema
- Service locator

Snapshot of whiteboard used during the discussion:



Service frameworks

We want:

- service management,
- service location,
- service configuration
- (and for it to not show up in the code - annotations are okay)
- we don't want to be forced to code in a certain way.
  Shortlist

- (JBoss) Seam - http://www.seamframework.org/
- Spring - http://www.springframework.org/
- Pico
  Also-rans:

- Plexus
- Apache HiveMind
- Google Guice
  Useful notes:

- Cocoon 2.2 uses Spring.
- Possibly consider having a static cover to locate services (e.g. Dspace.getService(name))

- - This should surely NOT be used within services NOR should it actually instantiate the services itself (locator only)

Funded work

- Data Model:

- Re-examine hierarchical identifiers
- Review versioning handling in model, implementation. ¬†Above and below item level?
- Propose implementation model of metadata at all levels
- Review Fedora & other repositories for commonalities and differences
- We should Re-evaluate the FRBR'ish data model in relation to SWAP -Mark Diggory
- Review Fedora & other repositories for commonalities and differences.
- Policy

- Re-evaluate policy management against new model. ¬†Ensure objects exist to apply policy against
- Services

- Group and refine service list from above.
- Evaluate Spring, Pico, and Seam
- Ensure SWAP occurs.
- UI Implications
- Migration / Upgrade Path

- Timing:

- Data Model, Policy and Services defined in first month+ ¬†(By Oct)

Extra Notes for people as out of the loop as Aaron:

Maven 2 site: http://projects.dspace.org/dspace/

Maven SNAPSHOT and Release repositories http://maven.dspace.org/

Source repository (SVN @ sourceforge): https://dspace.svn.sourceforge.net/svnroot/dspace/trunk/dspace

Issue tracker (sourceforge): http://sourceforge.net/tracker/?group_id=19984

Continuous Integration (Bamboo @ Johns Hopkins):

- https://integration.nts.jhu.edu/bamboo/browse/DS
- http://wiki.dspace.org/index.php/Emetsger:ContinuousIntegration
  Wiki (mediawiki): http://wiki.dspace.org/index.php/Main+Page

Note: The maven 2 site should be updated to have this information on it for 2.0 (some is here)

June 25 Funded Work Meeting

Agenda

- Milestones, interfaces, integration points
- Migration strategies
- Communications plan and initial work assignments

Discussion

- Data Model:

- Re-examine hierarchical identifiers
- Review versioning handling in model, implementation. Above and below item level?
- Propose implementation model of metadata at all levels
- Review Fedora & other repositories for commonalities and differences
- Policy

- Re-evaluate policy management against new model.
- Ensure objects exist to apply policy against
- Services

- Group and refine service list from above.
- Evaluate Spring, Pico, and Seam
- Ensure SWAP occurs.
- UI Implications
- Migration / Upgrade Path
- Fix DC namespace mappings per MDiggory work
- Timing:

- Data Model, Policy and Services defined in first month+ (By Oct)
- Integration and support of the Scholarly works application, ‚ÄúSWAP‚Äù

- Key use case: Support namespaces in metadata that can be attached to any container manifestation (SWAP and others)
- Also: correct existing DC namespace
- Ability to customize and ingest metadata supporting other DC application profiles in support of more complex items such as images, video and geospatial data

- Note: More than DC to handle here.
  - Considered straightforward; maintain existing capabilities. Direct MODS schema support?
- Possible metadata API adjustments.
- Integration and support for ‚ÄúSWORD‚Äù, a web service funded by JISC to allow for a common deposit interface

- Migration, considered straightforward.
- Existing SWORD, LNI, XMLAPI, other items from trunk will need to be supported.
- Ease of integration with elearning platforms such as Sakai, Moodle and Blackboard

- Cambridge: Existing LNI pulling items from DSpace no ingest.
  - Success at services layer, plus existing protocol implementations should address this.
- SWORD2?
- (provide) Backend data management and preservation for publishing systems such as Open Journal System (OJS)
  - See http://pilot.apsr.edu.au/wiki/index.php/OJS/OCS_Repository_Deposit_Project
  - Discussion: Do we need to provide more explicit services to enhance this type of work and avoid deep dependencies on Dspace internals? Component architecture TBD? Solvable via 2.0 services model?
- Improved support of SOA approach through further development of LNI(Light network interface), standard protocols to move data into and out of the DSpace repository.RESTful interface, entity system?

- Ingest, import, export provided via assortment of protocols (see existing packaging frameworks, crosswalks).
- Ability to customize the interface and workflow engine to better suit your institutions needs

- Predates Manakin? Underlying requirements are?
  - Note: I18n strategies diverge between xmlui and jspui. Question of degree to which model work influences this - UI work out of scope beyond support for new model?
- Workflow is a separate issue, although services interfaces should be compatible with pluggable workflow
- Ability to handle more complex objects with ease, such as websites, multiple versions of documents etc. Using the FRBR model.

- Part of new Data model.

How to work

- Call for TDD, using mvn, spring, etc.
- Do we use new packages? (yes)
- Anemic model rather that rich class model.
- Model classes versus interfaces.
- DAOs on in memory databases versus mocks.
- (Discussion about integrity checks)
- Events - currently synchronous
- Discussion of the current context, and how to revise for service model. Pass around GUIDs internally?
- Mention of removing filename from external URLs (use disposition in browser headers)
- Communities / Collections -> Metadata (logos, etc) as items? as metadata? How preserve?
- Everything is an Item? (Collections are Items with policy attached)

How we work:

- Work out of a branch of the SF SVN parallel directories ¬†(? 2.0.0-SNAPSHOT)
- Normal dev comms on the mailing lists and IRC.
- Weekly Skype (or conf call) checkin for team (Mondays?) starting Sep.
- Mark: 70% starts Sept.
- Aaron: ? 80% starts Sept, slightly before.
- Jim: 50% starts Sept
- Graham: ? 75% x 3 mo, starts ??
- Using Jira for tasks.

- Or possibly Trac. foundation is investigating hosting ‚Äì Brad McLean 07/08/2008

Task & Milestone outline

- Task: Define design / coding practices.
- Task: Prepare initial API examples, test case examples, database construction examples, unit test environment, build environment.
- Task: Design replacement mechanism to eliminate Context passing.
- Task: Define proposed final data model, object model
- Task: Enumerate service APIs
- Milestone: Design complete
- Tasks: Review APIs and write tests
- Tasks: Implement APIs (transfer code)
- Tasks: Refactor UI / Exports
- Milestone: Alpha
- Tasks: UI Integration
- Tasks: Develop migration support tools
- Task: Collate Documentation
- Tasks: Service fixes
- Milestone: Beta
- Task: Documentation updates
- Milestone: Release