DSpace Summer of Code Ideas 2012

Ideas for Potential DSpace Summer of Code 2012 projects

Add your ideas here!

⚠

Please add your suggestions for GSoC 2012 projects related to DSpace! If you are interested in mentoring, please let us know! Also, be sure to visit the listing of Past GSoC Project Ideas below, to see if anything there is still relevant.

Please add your own ideas to the table below, and feel free to volunteer as a mentor for any existing idea

Summary	Relevant DSpace component (s)	Detailed Description	Mentor volunteer (s)
Stop Using Email as User Identifier	EPerson Authentication Manager	Use of email address as a persistent identifier for the DSpace conflicts with the fact that email addresses are not persistent. Email addresses go away and/or are reassigned to other individuals. There are also policy concerns with Authenticators like Shibboleth and CAS that may or may not deliver an email address as a organizational policy. This Task is a placeholder to identify a solution to correct for the problem. 1.) DSpace should use a different identifier / key for the EPerson (netid? or combination of "authenticator + netID") 2.) DSpace should make providing an email address as optional for cases where the Authentication features lack this specific capability. 3.) Issuing emails should be optional for accounts without email addresses. 4.) Stop storing email address (or any other detail about who made the change) in dc.description.provenance field. One proposed solution to this problem is that the Authentication Method should be broken off of EPerson and stored separately, making EPerson a "Profile" and the method of Authentication be stored separately (Password, Certificate, LDAP, Shibboleth, CAS, Facebook, Google) Different AuthenticationMethods may store the data as they see fit. And the Profile would store only those local details for that user. See for more detail: https://jira.duraspace.org/browse/DS-937?focusedCommentId=24361#comment-24361	
GFDAO (Generic Fedora Data Access Objects)	storage-triplestore storage-fedora dspace-storage	Bring together the storage-fedora and storage-triplestore projects to produce a Fedora store based persistence tier that includes use of SPARQL to retrieve "Related and/or Contained" Fedora Objects. Objective is: That once DSpace Communties, Collections, and Items are represented as Fedora Objects with relationships captured in RELS-EXT, then a data persistence and mapping layer will need to be engineered for DSpace / Fedora interaction. Such a Data Access tier will provide a suite of Domain Model Centric Data Access Objects that will provide a DOFM (DSpace Object Fedora Mapping). Semantic Queries will be utilized to acquire sub collections, sub-communities, Items and parent objects. Further thought can be placed into creating a generic Fedora Data Access Object that is capable of resolving any RELS-EXT (or other rdf based relation for that matter).	Mark Diggory
Disseminator Framework	ItemView Item Bitstream XMLUI	A Disseminator Framework will associate Disseminators with Items and/or Bitstreams, Disseminators will combine METS file description and behavior sections to supply the user interface with a standard representation of the dissemination services that can be applied to a content bitstream in DSpace. See for further background: • http://www.loc.gov/standards/mets/METSOverview.v2.html • https://wiki.duraspace.org/display/DSPACE/DSpaceMETSSIPProfile • http://scm.dspace.org/svn/repo/dspace/trunk/dspace-xmlui/dspace-xmlui-api/src/main/java/org/dspace/app/xmlui/objectmanager/ • The XMLUI XSLT templates can be extended to support the definition of different types of METS "Behaviors" (links, embedded, alternative protocols, external embedded sites) Disseminator Viewers and the Framework will provide a "Pluggable" solution for DSpace Addon Modules to Deliver new Viewer capabilities into the DSpace UI	Mark Diggory

Extend Metadata Framework to Support Stronger Typing and Validation	MetadataValue MetadataField MetadataSchema Community Collection Item Bitstream XMLUI	Extend the DSpace Metadata Domain Model to support the following features: • Community, Collection and Bitstream Level Metadata • Create "ContentType" or "Classes" Domain Model for Items and Bitstreams that define Metadata Fields that Should / Could be Present and their occurence • Allow MetadataFields to define the appropriate Authority Controls that can be applied to them at the Domain Model level so that they can be used to inform Submission and Edit Forms (rather than using dspace.cfg properties. • Correct Dublin Core Model and Add More default namespaces (Dublin Core Terms, ETD, VRACore and BIBO Namespaces Research Applying METS more fully in the DSpace Domain Model • Allow Metadata "Files" that are stored in Bitstreams attached to Item but rendered in XMLUI and Descriptive, Administrative Metadata Sections • Allow Behaviors to be attached to Fiel Descriptions, Structure Maps, etc • Allow Structure Maps to be more descriptive of the contents of the item. • Describe and attach Bitstream level metadata as descriptive and administrative metadata sections • Describe and Attach Collection and or other relationships in administrative metadata sections (Similar to Fedora) • Formalize how "Relations" Between Items should be expressed in METS metadata (see Fedora Rels-Ext for examples). Create a Validation Service Capable of Reviewing the Metadata and Content of the Items, Create Curation Tasks that can call the Validation Service by the Repository Managers	Mark Diggory
Rewrite Packager Framework	dspace-api packagers crosswalks	Refactor Packagers to support Chain of Command The packager framework in DSpace is rather rigid and unwheldy, an excellent project would refactor the Packager and Crosswalk frameworks to support more basic packagers and crosswalks. DSPace METS SIP Packages and/or SWORD packages are not the only types of packages out there. in fact, METS was never meant to be "repository specific", but to be "content specific". A better Packager Framework would gracefully degrade based on features it could detect about the incoming content being ingested, even in the METS case, DSpace should be "Profile Agnostic" and accept any METS package to derive one or more DSpace items.	Mark Diggory
DSpace Core Domain Model	dspace-api dspace-services dspace-core	Domain Model The Domain Model Project seeks to resurrect the DAO prototype work of James Rutherford and others to provide a cleaner mean to replace the underlying data storage tier in DSpace. Such efforts will more than likely be pivotal in achieving a DSpace with Fedora inside integration. Likewise, anyone interested in providing support for additional database vendors will find the DAO refactoring project of value in terms of providing a separation of relations data storage away from the Domain and Business Services tier of the DSpace application.	Mark Diggory
Accessibility	dspace-api dspace-xmlui	DSpace is selected as an institutional repository platform for one special library for the blind (details come later). Its implementation/adaptation for institutional use is currently in progress (near end-stage). The institution would be willing to share some ideas, requirements and code already done in order to make solution widely available for institutions of similar type. Some of important aspects and ideas of customizing DSpace for such purposes would be: • adopt it on such way that it provides higher usability for blind persons • create mobile application (Android, IOS...) which could provide blind users possibility to browse repository, download and listen books • provide member management interface for closed-type library repository • adopt DSpace components in order to better fit institutional needs of such libraries • provide better support for delivering secured content (DRM of PDF documents, Daisy books etc)	Bojan Suzic
Enhance REST-API Implementation	REST API	Update the REST API to use a more common framework (like Spring WebMVC or Jersey), rather than the current Sakai bus implementation. (There could be other REST API projecst we could add to this list – the goal being to continue to move the REST API forward & stabilize it even further.)	*Needs mentor(s)*
Metadata Reconciliation with Authoritative Sources	EditItemStatus EditItemMetadata	Build a tool that allows curators to compare DSpace metadata with metadata from authoritative sources. The tool will allow curators to see DSpace metadata alongside metadata from a system such as CrossRef or PubMed. Individual metadata fields will be color-coded according to the degree of consistency. Curators will be able to click a button for each metadata field they wish to import from the authoritative source. An initial design for such a tool was created during a hackathon at the 2011 Code4Lib conference. This design, called HAMR, provides a good starting point for the project, but there is considerable room for a student to work with the DSpace community to determine the final design of the tool.	Ryan Scherle
Enhancements to DSpace Statistical Reports, including one or more of the following: Develop Visualizations, tabbed/paginated interface, and/or "Export to CSV"	Solr Statistics Engine XMLUI	The current DSpace Solr Statistics Engine interface is rather simplistic in nature. Maybe it's time to spice it up a bit and add some better visualizations (via something like Google Charts API). It also could be worthwhile to paginate or tab the interface, both to help improve performance (less queries on one page = better performance) and so that administrators are first shown a "general summary" page, but can choose to visit other pages/tabs to get more detailed statistics in the area of interest. • Another idea is to support "Export to CSV/Excel" for a date-range, allowing users to generate Excel reports based on statistical information. • Related: recent Ohio State Statistical "Report Generator" improvements: Screenshot (Zoomable) & Stats Code	*Needs mentor(s)*