

DSpace Summer of Code Ideas 2011

This is an archived listing of project ideas for Google Summer of Code in 2011. To see which projects actually took part in GSoC 2011, please visit the [GSoC 2011 Projects](#) page

Contents

- 1 [Ideas for Potential DSpace Summer of Code 2011 projects](#)
- 2 [Past DSpace GSoC Projects](#)
- 3 [Past Ideas for Potential DSpace GSoC projects](#)

Ideas for Potential DSpace Summer of Code 2011 projects

Summary	Relevant DSpace component (s)	Detailed Description	Mentor volunteer (s)
GFDAO (Generic Fedora Data Access Objects)	storage-tripstore storage-fedora dspace-storage	Bring together the storage-fedora and storage-tripstore 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 Communities, 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: <ul style="list-style-type: none"> http://www.loc.gov/standards/mets/METSOOverview.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/ <p>The XMLUI XSLT templates can be extended to support the definition of different types of METS "Behaviors" (links, embedded, alternative protocols, external embedded sites)</p> <p>Disseminator Viewers and the Framework will provide a "Pluggable" solution for DSpace Addon Modules to Deliver new Viewer capabilities into the DSpace UI</p>	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: <ul style="list-style-type: none"> 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 occurrence 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 Field 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 unwieldy, 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
Enhanced RESTful API	dspace-api dspace-services rest	Building on previous years' REST projects, this project is to result in: <ul style="list-style-type: none"> (non-basic!) authN and authZ functionality write / admin methods minimum of JSON and XML output formats communication and collaboration with community, on: <ul style="list-style-type: none"> testing, benchmarking, hacking demo apps promotion and awareness documentation <p>Note: The student who previously implemented RESTful API (both write and read-only) project has indicated he would like to continue work on it outside GSoC, so this entry should perhaps be removed for now</p>	
Usage statistics reports	dspace-xmlui dspace-jspui? dspace-stats? dspace-solr?	The new solr statistics engine in DSpace needs to generate more reports, replace the legacy "site-wide" statistics report that is still in use, and make use of Google Charts API (or similar) to make graphs easier to read, compare between items, etc. This project will result in <ul style="list-style-type: none"> Legacy site-wide statistics (generated w/ log analyzers) replaced with reports generated from new statistics engine More usage event types being captured (?) Reports presenting aggregations of counts across site/communities/collections/items/etc 	Kim Shepherd Andrea Schweer

RTMP A/V streaming integration	dspace-api dspace-xmlui dspace-jspui?	<p>Integrate DSpace authentication, authorization and bitstream dissemination with an open source RTMP streaming server such as RED5. Custom flash player is also desirable.</p> <p>This project will result in</p> <ul style="list-style-type: none"> Ability for DSpace to serve most A/V content via RTMP <ul style="list-style-type: none"> If possible, this project should also allow DSpace to stream content from DuraCloud's Media Streaming Service which supports RTMP Ability to deny playback for non-public bitstreams to unauthorized users Custom flash player for use with XMLUI and JSPUI(?) Integration with UsageEvents system to fire events on pause, resume, connect, etc. 	Kim Shepherd Andrea Schweer
New UI built over DSpace API, based on popular frameworks		<p>Currently, DSpace functionality could be accessed using JSP and XML user interfaces. As in the meantime many other technologies arrived which could provide different way of user experience and UI customization, the idea is to create a new interfaces using some of the following technologies:</p> <ul style="list-style-type: none"> Vaadin - component based framework, server run, provides rich visual experience to the users. Demonstration of functionality could be accessed here. GWT or ExtGWT based frameworks. The bridge between DSpace API and client run components from these frameworks should be developed and tested. Demonstrations are available here and here. Flex, different approach by Adobe. Provider and bridge between DSpace API and Flex client application should be provided. <p>The most important functionality these interfaces should deliver is user interface which has attractive look and feel and provides users with enhanced usability. Additionally, easier customization of resulting interface would be additionally appreciated. As of project complexity and project deadlines, support for many administrative functions in the interface is not so important in current stage. It is more important to have rich and attractive user interface conforming to actual visual and usability trends.</p>	Bojan Suzic
Accessibility	dspace-api dspace-xmlui	<p>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.</p> <p>Some of important aspects and ideas of customizing DSpace for such purposes would be:</p> <ul style="list-style-type: none"> 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
Scriptable Curation Tasks	dspace-xmlui dspace-jspui curation tasks	<ul style="list-style-type: none"> DSpace's Curation System allows administrators to run tasks on individual items or sets of items at various stages of the item's lifecycle. At the moment, new tasks need to be written in Java, compiled and made available to the DSpace codebase. The curation system could be extended by allowing administrators to write curation tasks in a simplified programming language (eg Groovy, Jython or JRuby) that can then be run in a one-off fashion, or stored within DSpace and re-run at a later point. This would give DSpace administrators a middle ground between bulk metadata editing and writing full-out Java code; it would also allow simple code to be run by people without shell access to the DSpace server. <p>Example tasks are:</p> <ul style="list-style-type: none"> For all items (in the repository/in a given collection), if the item's dc.description metadata contains a given string, move the item's bitstream from the ORIGINAL bundle to a different bundle For all items (in the repository/in a given collection), if the item's metadata value in dc.identifier.doi starts with 'http://xyz/', strip out that part and write the remainder of the value back into dc.identifier.doi For all items (in the repository/in a given collection), if the item's metadata value in dc.identifier starts with 'ISBN: ' or 'doi: ', remove the value, strip out that part and write the remainder of the value into dc.identifier.isbn or dc.identifier.doi as appropriate <p>The first task cannot be done using bulk metadata editing because it affects the item's bitstreams as well as the metadata. The other two tasks could be done using bulk metadata editing but, for someone who can write code, may be faster to do in a scripting language than via metadata editing.</p> <ul style="list-style-type: none"> This project would consist of writing the infrastructure that allows administrators to define and run these tasks. <p>A good trade-off must be made in exposing enough of DSpace's data model to be useful while hiding some of the more complex parts; some precautions will most likely need to be taken to ensure that the core DSpace functionality can't be compromised by the tasks.</p>	Andrea Schweer Kim Shepherd
Improve Submitter User Experience	dspace-xmlui dspace-jspui? webmvc?	<p>Generally clean up submission process, especially at UI level, and provide a more user friendly submitter experience. (Need to brainstorm out exactly what changes to make obviously)</p> <p>Examples may include:</p> <ul style="list-style-type: none"> Finding ways to ease interaction with the submission UI via jQuery or similar technologies (improved step-by-step feedback, improved upload process, etc) Provide popup help/tips on various submission fields? Perform Usability Studies on users of the system to find pain points during submission. (i.e. does user know what collection they are submitting to, can they find the item they just submitted, do they have to delete everything in an entry to change an author's name while still preserving the order of authors, do all the checkboxes and radio bubbles make sense as to what they do?) Finding ways to minimize the length of time required to submit something. Since XMLUI is the main interface for dspace, it would be required that the project primarily improves the XMLUI interface. Improving JSPUI, SWORD/easy-deposit, WebMVC, or batch submission could be additional improvements to work on. (Other thoughts/ideas welcome...) <p>Comment (Andrea): I'd really like to see an option for DSpace administrators to configure the submission UI via the web interface. I'm aware of the Google Summer of Code 2009 Submission Enhancements but I don't know what the outcome of this project was -- it doesn't seem to have made it into DSpace yet.</p>	Peter Dietz Andrea Schweer (as co-mentor)

Past DSpace GSoC Projects

Please visit [Past DSpace Summer of Code Projects](#) for a full listing of all past DSpace GSoC projects.

Past Ideas for Potential DSpace GSoC projects

We have archives of all our Past GSoC Ideas Pages still available for reference/ideas. *However, you should check with the available mentors before suggesting any of these older project descriptions. In many cases these projects may no longer be applicable, or may require rethinking to bring them up to date.*

- [DSpace Summer of Code Ideas 2012](#)
- [DSpace Summer of Code Ideas 2011](#)
- [DSpace Summer of Code Ideas 2010](#)
- [DSpace Summer of Code Ideas 2009](#)
- [DSpace Summer of Code Ideas 2008](#)