What is DSpace?

- Store Content
- Manage Content
- Distribute Content
Model & Concepts
DSpace data model
DSpace data model
Practical

Let us pause at this point and ...

- Look at the DSpace user interface
- Investigate the data model
DSpace Architecture
Item lifecycle

- Ingest
  - Batch import
  - Web-based submission
- Published
  - Accessible over the web
- Withdrawn
  - Hidden from view
- Expunge
  - Permanently removed
Bitstreams & formats

- Bitstream formats
  - MIME types
  - File extensions
- Support levels
  - Supported
  - Known
  - Unsupported
Registry

- Metadata Registry
  - Approved fields
  - Multiple ‘schemas’

- Format Registry
  - Approved file formats
  - Support levels
  - Browser support
Handles

- Persistent identifier
- Communities
- Collections
- Items

Format:

hdl:1721.123/4567
http://hdl.handle.net/1721.123/4567
- prefix
- identifier
E-People & groups

- Individual users
- Authenticated
  - Password
  - LDAP
  - Shibboleth
  - others...
- Groups of E-People
Authorizations

- **Role**
  - Several actions
  - Upon a single object
  - Responsibility

- **Privilege**
  - Individual actions
  - Upon a single object
  - Grant the ability to perform a specific action over an object
**Community:**
ADD / REMOVE

**Collection:**
ADD / REMOVE
DEFAULT_ITEM_READ
DEFAULT_BITSTREAM_READ
COLLECTION_ADMIN
WORKFLOW_STEP_1
WORKFLOW_STEP_2
WORKFLOW_STEP_3

**Item:**
ADD / REMOVE / READ / WRITE

**Bundle:**
ADD / REMOVE

**Bitstream:**
READ / WRITE

**Authorizations**
Crosswalks

- Export metadata into various formats

- Current crosswalks:
  - DIM
  - MODS
  - DC & QDC
  - Custom XSLT
OAI-PMH

- Open Archive Initiative’s Protocol for Metadata Harvesting
- Export metadata
- Uses crosswalk plugins
- Google scholar
- Oaister
Media filters

- Plugins for file type support

- Current filters
  - Adobe PDF
  - HTML, Text
  - Microsoft Word
  - JPEG
  - Branded Preview
Building DSpace
Configuring DSpace
Topics

- The Basics
- Authentication
- Search & Browse
- Plugins
- XMLUI
The basics

- Java Property file
- name = value
- Use slash ("/") to continue to the next line
Browse

- User exploration
- Four fixed browse indexes
  - Title
  - Subject
  - Author
  - Date
Default DSpace browse indexes:

webui.browse.index.date     = dc.date.issued
webui.browse.index.author   = dc.contributor.*
webui.browse.index.title    = dc.title
webui.browse.index.subject  = dc.subject.*
Search

- Targeted retrieval
- Unlimited number of search indexes
- Manakin tweak to add index labels
Default DSpace search indexes:

```
search.index.1  =  author:dc.contributor.*
search.index.2  =  author:dc.creator.*
search.index.3  =  title:dc.title.*
search.index.4  =  keyword:dc.subject.*
search.index.5  =  abstract:dc.description.abstract
search.index.6  =  author:dc.description.statementofresponsibility
search.index.7  =  series:dc.relation.ispartofseries
search.index.8  =  abstract:dc.description.tableofcontents
search.index.9  =  mime:dc.format.mimetype
search.index.10 =  sponsor:dc.description.sponsorship
search.index.11 =  identifier:dc.identifier.*
search.index.12 =  language:dc.language.iso
```
texas
Search for ‘texas’ in any field or extracted text

(texas)
Same as above

(title:texas)
Search for ‘texas’ in the title

(title:texas) AND (author:john)
Search for items with ‘texas’ in the title and john as an author

(title:texas) AND ((author:john) OR (author:leggett))
Search for items with ‘texas’ in the title and either john or leggett as an author
What is Manakin?

- Interface framework
  - Modular
  - Extendable
  - Tiered

- A new interface to DSpace
What can Manakin do?

- Modify Look-and-Feel
- Brand Content
- Visualize Metadata
- Provide Modularity
Tiers

1. Style Tier
   - Create simple themes
   - XHTML + CSS

2. Theme Tier
   - Create complex themes
   - XSL + XHTML + CSS

3. Aspect Tier
   - Add new features
   - Cocoon + Java
Apache Cocoon

- Web development framework
- Pipeline-based architecture
- SAX-based XML
- Modular components
  - Generators
  - Transformers
  - Serializers
Cocoon’s Pipeline Model

Sitemap defines the pipeline

Generator

Transformer X

Transformer Y

Serializer

Cocoon Pipeline

<doc>
  <z/>
</doc>

<doc>
  <x/>
</doc>

<doc>
  <x/>
  <y/>
</doc>
Three components

- Aspects
- Themes
- DRI Schema
Structural overview

Content Generation

Aspect Chain

Aspect 1

Aspect 2

Aspect 3

Style Application

Themes

Theme A

Theme B

Theme C

DRI Document
DRI Schema

Abstract representation of a repository page

- Digital Repository Interface

- Metadata elements
  - METS, MODS, QDC, DIM
  - Extendable to other metadata formats

- Structural elements
  - TEI (light)
DRI Schema

Abstract representation of a repository page

Based upon TEI

Native metadata formats: METS, MODS, DIM...
Aspects

Implement a set of related features

- Applied to all pages
- Interact with the repository
- “Aspect Chain”
  - Input DRI
  - Output DRI
Aspects

Implement a set of related features

Core Aspects:
- Artifact Browser
- E-Person
- Submission
- Administrative

Possibilities:
- Shopping cart
- Specialized searching
- Customized workflow
Aspects

Implement a set of related features

- Self contained packages
  - Java source code
  - Javascript flow scripts
  - XML Configuration
Themes

Stylize content into a particular look-and-feel

- Defines how the repository looks
- May apply to:
  - A single page
  - All pages in a collection
  - All pages in a community
  - The whole repository
Themes

Stylize content into a particular look-and-feel

- Self contained packages
  - XSL stylesheets
  - CSS stylesheets
  - Images
  - Static resources
Putting it all together

Content Generation

Artifact Browser → Eperson → Submission

Style Application

Theme B

<document>
  <body/>
  <options/>
  <meta/>
</document>
Putting it all together

Content Generation

Artifact Browser → Eperson → Submission

Style Application

Theme B

<document>
  <body>
    <div>
      <head>Viewing an item</head>
      <includeSet>
        <objectInclude source="hdl:12345/23"/>
      </includeSet>
    </div>
  </body>
  <options/>
  <meta>
    ...
    <objectMeta>
      <object identifier="hdl:12345/23"/>
      .... METS document for item ...
    </object>
    </objectMeta>
  </meta>
</document>
Putting it all together
Putting it all together

XML Document:

```xml
<document>
  <body> ... </body>
  <options> ...
    <list n="account">
      <item xref="/logout"> logout </item>
      <item xref="/profile"> Profile </item>
      <item xref="/submissions"> Submissions </item>
    </list>
  </options>
  <meta> ...
    <userMeta> ... </userMeta>
    <objectMeta>
      <object identifier="hdl:12345/23"/>
      .... METS document for item ...
    </object>
  </meta>
</document>
```
<html>
<head> ... </head>
<body>
<div>
<h1>viewing an item</h1>
<table>
... The item's metadata ...
</table>
</div>
<li><a href="/logout">Logout</a></li>
<li><a href="/profile">Profile</a></li>
<li><a href="/submissions">Submissions</a></li>
</div>
</body>
</html>
1. **Style Tier**
   - XHTML + CSS
   - Create simple themes

2. **Theme Tier**
   - XSL + XHTML + CSS
   - Create complex themes

3. **Aspect Tier**
   - Cocoon + Java
   - Add new features
Create a theme

Getting Started
1. Create a theme
2. Install your theme
3. Style with CSS
4. Overview
Copy Theme Template

```
cd [tomcat]/webapps/xmlui/themes
mkdir [your theme]
cp -r /class/shared/template/* [your theme directory]
cd [your theme directory]
mv template.xsl [your theme directory].xsl
```

Create a theme
Sitemap Path & Name

<map:component-configurations>
  <global-variables>
    <theme-path>[Your theme directory]</theme-path>
    <theme-name>The [Your theme name] template</theme-name>
  </global-variables>
</map:component-configurations>

Note: The theme path is the directory name of your theme. The theme name is the name you assign the theme and only used to describe the theme.
Sitemap Transformation

<!-- Step3: Transformation to XHTML-->

<map:transform src=”[Your theme name].xsl”/>

Note: The XSL document allows for more custom themes during the transformation of the DRI (Direct Rendering Infrastructure) document to XHTML.
Install a theme

Register your theme
1. Edit Manakin config file
2. Name your theme
3. Attach your theme to a handle or regular expression
4. Save changes and restart Tomcat
Install a theme

Edit Manakin config file

Navigate to and open config file.

1. cd [press the Enter key]

2. cd dspace/config

4. pico xmlui.xconf
Register theme: edit Manakin config file

Register a theme as the main theme
<themes>
  <theme name="[Your theme name as registered in theme sitemap]"
    regex=".*"
    path="[Your theme path as registered in theme sitemap]/ " />
</themes>

Register a theme as a handle theme
<themes>
  <theme name="[Your theme name as registered in theme sitemap]"
    handle="123456789/2"
    path="[Your theme path as registered in theme sitemap]/ " />
</themes>
Install a theme

Edit Manakin config file

1. Save changes: “Ctrl O” to save in Pico

2. Restart Tomcat
   - [tomcat]/bin/shutdown.sh
   - [tomcat]/bin/startup.sh

3. Your theme is installed

4. Save your theme to source!
Styling with CSS

- Location of CSS files
- Register CSS in Sitemap and target browsers
- Use the CSS declarations already in the DRI Document
Edit/Maintain CSS files

1. CSS files are stored in [your theme directory]/lib folder.

2. Create new CSS files for specific browsers.

3. Add only the differences to browser specific CSS files.
Styling with CSS

<!-- Step 1: Generate the DRI page -->
<generate type="file" src="cocoon://DRI/{1}"/>

<!-- Step 2: Add page metadata -->
<select type="browser">
  <when test="explorer7">
    <transform type="IncludePageMeta">
      <parameter name="stylesheet.screen#1" value="style.css"/>
      <parameter name="stylesheet.screen#2" value="style-ie.css"/>

      <parameter name="theme.path" value="{global:theme-path}"/>
      <parameter name="theme.name" value="{global:theme-name}"/>
    </transform>
  </when>
  <otherwise>
    <transform type="IncludePageMeta">
      <parameter name="stylesheet.screen" value="style.css"/>

      <parameter name="theme.path" value="{global:theme-path}"/>
      <parameter name="theme.name" value="{global:theme-name}"/>
    </transform>
  </otherwise>
</select>
Common browsers to target

```html
<!-- Step 2: Add page metadata -->
<map:select type="browser">
  <map:when test="explorer6">
    ...
  </map:when>
  <map:when test="explorer7">
    ...
  </map:when>
  <map:when test="explorer8">
    ...
  </map:when>
  <map:when test="safari2">
    ...
  </map:when>
  <map:when test="safari3">
    ...
  </map:when>
  <map:otherwise>...</map:otherwise>
</map:select>
```
Common DRI CSS Declarations

1. ds-main: main wrapper around all declarations
2. ds-header: Page header including logo, title
3. ds-trail: Page breadcrumb
4. ds-user-box: User access and profile
5. ds-body: Main content
6. ds-options: Navigation
7. ds-footer: Footer
Theme overview

Components

- Sitemap
  - The glue that binds
  - Reference components
  - Is configurable

- XSL Document
  - Transforms DRI to HTML

- CSS
  - Styles the resulting HTML
Theme overview
Practical

Let us pause at this point and ... 

- Stylize our theme using CSS
Introduction to XSL
What is XSL?

- eXtensible Stylesheet Language
- Transform XML documents into new XML documents
- XSL is encoded in XML
- XSL officially adopted by W3C
XSL architecture

- Blue = XML Document
- Orange = Software

Changing the stylesheet will produce a different target document
Templates

- Match patterns
  - Until all source elements are processed
- Produce XML fragments
- All templates together produce the target XML document
Example 1 (architecture)

Source Document (DRI)

```xml
<figure source="/images/logo.png">
    My university logo
</figure>
```

XSL Stylesheet (XSL)

```xml
<xsl:template match="figure">
    <img src="{@source}" alt="{text()}"/>
</xsl:template>
```

Target Document (HTML)

```html
<img src="/images/logo.png" alt="My university logo"/>
```
Example 2 (templates)

XSL Stylesheet (XSL)

```xml
<xsl:template match="figure">

  <xsl:if test="@target">
    <a href="{@target}"
       ><img src="{@source}" alt="{text()}/>
  </a>
  </xsl:if>

  <xsl:if test="not(@target)"
  ><img src="{@source}" alt="{text()}/>
  </xsl:if>

</xsl:template>
```

Target case, image with a hyperlink

Normal case, just an image
Example 2 (templates)

Source Document (DRI)

```html
<figure source="/images/logo.png" target="http://my.university.edu/">
    My university logo
</figure>
```

Target Document (HTML)

```html
<a href="http://my.university.edu/">
    <img src="/images/logo.png" alt="My university logo"/>
</a>
```
Template selection

- Each template handles a specific element
- Match rules used to determine which template is applied
- `<apply-templates/>` finds the next template to match
XSL Stylesheet (XSL)

```xml
<xsl:template match="hi">
  <span class="{@rend}">
    <xsl:apply-templates/>
  </span>
</xsl:template>

<xsl:template match="xref">
  <a href="{@target}">
    <xsl:apply-templates/>
  </a>
</xsl:template>
```

Template for highlights: bold, italics, underline, etc...

Template for hyperlinks

Example 2 (selection)
Example 2 (selection)

Source Document  (DRI)

```html
<hi rend="bold">
  <xref target="http://my.university.edu/">
    Click me to go to my university home page
  </xref>
</hi>
```

Target Document  (HTML)

```html
<span class="bold">
  <a href="http://my.university.edu/">
    Click me to go to my university home page
  </a>
</span>
```
XPath

- Used for Addressing part of an XML document
  - navigate XML structure
- Path is relative to the root or current node
  - /document/body/div/head
  - div/head
Welcome to my digital repository. Here you can browse the repository for interesting items.

Browse by:
- Titles
- Authors
- Subjects
- Dates
Welcome to my digital repository. Here you can browse the repository for interesting items.

Browse by:
- Titles
- Authors
- Subjects
- Dates
<body>
  <div>
    <head>Welcome</head>
    <p>Welcome to my digital repository.</p>
    <p>Here you can browse the repository for interesting items.</p>
  </div>
  <div>
    <head>Browse by:</head>
    <list>
      <item>Titles</item>
      <item>Authors</item>
      <item>Subjects</item>
      <item>Dates</item>
    </list>
  </div>
</body>
Welcome to my digital repository.
Here you can browse the repository for interesting items.

Browse by:
- Titles
- Authors
- Subjects
- Dates
Welcome to my digital repository. Here you can browse the repository for interesting items.

Browse by:
- Titles
- Authors
- Subjects
- Dates
<body>
  <div>
    <head>Welcome</head>
    <p>Welcome to my digital repository.</p>
    <p>Here you can browse the repository for interesting items.</p>
  </div>
  <div>
    <head>Browse by:</head>
    <list>
      <item>Titles</item>
      <item>Authors</item>
      <item>Subjects</item>
      <item>Dates</item>
    </list>
  </div>
</body>
Welcome to my digital repository.
Here you can browse the repository for interesting items.

Browse by:
- Titles
- Authors
- Subjects
- Dates
Replace home page welcome text.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns="http://di.tamu.edu/DRI/1.0/" xmlns:i18n="http://apache.org/cocoon/i18n/2.1" version="1.1">
  <body>
    <div id="file.news.div.news" n="news" rend="primary">
      <head>Introduction</head>
      <p>The Repository's purpose is to collect, record, provide access to, and archive the scholarly and research works that reflect the intellectual and service environment of the campus.</p>
    </div>
  </body>
</document>
```

```xml
<xsl:template match="//dri:body/dri:div[@n='news']/dri:p[1]">
  <text>Your new content</text>
</xsl:template>
```
Replace home page search box.

Replace home page search box.

<Xsl:Template match="//dri:body/dri:div[@n='front-page-search']">
  <text></text>
</Xsl:Template>
Let us pause at this point and ....

- Run an XSL template
- Experiment with XPath
Theme Tier
1. **Style Tier**
   - XHTML + CSS
   - Create simple themes

2. **Theme Tier**
   - XSL + XHTML + CSS
   - Create complex themes

3. **Aspect Tier**
   - Cocoon + Java
   - Add new features
XSL Library

- **Structural**
  - Main structural components of theme

- **DIM-Handler**
  - Item, Collection, and Community display types

- **General-Handler**
  - Information about the file or METS
## Base XSL Library

<table>
<thead>
<tr>
<th>Package</th>
<th>Structural display</th>
<th>Metadata handlers</th>
<th>General handlers</th>
</tr>
</thead>
<tbody>
<tr>
<td>/themes/dri2xhtml.xsl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/themes/dri2xhtml/structural.xsl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/themes/dri2xhtml/DIM-Handler.xsl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/themes/dri2xhtml/General-Handler.xsl</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Your Theme

DRI Document

Sitemap

XSL

CSS

XHTML Document

Base XSL Library

i18n

Default Catalogue

Complete picture
Structural XSL

- Contains
  - dri:document
  - buildHead
  - buildHeader
  - buildFooter
  - dri:trail
  - dri:body
  - dri:options
  - many others (ex:dri:div, dri:table, dri:p)
Copy template to [your theme].xsl file for customization.

<xsl:template name="buildFooter">

<!-- Customize footer -->

</xsl:template>
Let us pause at this point and ...

- Customize our theme’s footer
Metadata handlers (DIM-Handler)

- Display items, collections, and communities
- Four display types
  1. SummaryList
  2. SummaryView
  3. DetailedList
  4. DetailedView
Copy select template to [your theme].xsl file for customization.

```xml
<xsl:template match="dim:dim" mode="itemSummaryView-DIM">
  <!--Add metadata field dc.subject to item summary-->
  <xsl:if test="dim:field[@element='subject']">
    ...
  </xsl:if>
</xsl:template>
```

dri2xhtml/DIM-handler.xsl - > [your theme].xsl
Let us pause at this point and ...

- Modify an item’s display
General Handler

- Bitstream metadata and link to bitstream
- Link to License
dri2xhtml/General-handler.xsl - > [your theme].xsl

Copy select template to [your theme].xsl file for customization.

```xml
<xsl:template match="mets:fileGrp[@USE='CONTENT']">
  <!--edit title and display of bitsreams-->

  ...

</xsl:if>
</xsl:template>
```
Let us pause at this point and ... 

- Modify a bitstream display
Advanced Topics
Ingest process

DSpace Ingest Process

- External SIP
- Batch Importer
- Web-based Submission
- In Progress Submission
- Workflow
- Item Installer
- Item in DSpace
Batch item import

- Command line import
- Directory structure
  - “contents” file
  - “dublin_core.xml” file
  - “handle” file
  - bitstreams files
Format:

```
<filename>
```

or

```
<filename> <tab> bundle:<bundle name>
```

Examples:

```
dissertation.pdf
mods.xml       bundle:METADATA
license.txt    bundle:LICENSE
```
Format:

```
-r -s <#> -f <filename> <tab> bundle:<bundle name>
```

Where:
- `-s <#>` Which ‘asset store’ to store the file
- `-f <filename>` The bitstream’s file name
- `bundle:<bundle name>` (Optional) Which bundle to place in

Examples:

```
-r -s 1 -f dissertation.pdf
-r -s 2 -f mods.xml  bundle:METADATA
-r -s 1 -f license.txt  bundle:LICENSE
```
./dsrun org.dspace.app.itemimport.ItemImport

-a          Add new items to DSpace
-c <coll>    Which collection to add them to
-e <email>  Existing user who is adding these items
-m <path>   Create a log file for this import
-s <path>   Location of the import files

-t          Do not run, just test the import for validity
-h          Print command line options and their
description
Examples

./dsrun org.dspace.app.itemimport.ItemImport -a
   -c 123456789/5 -e scott@library.tamu.edu
   -m /path/to/file.map -s /path/to/import

-a Add new items to DSpace
-c <coll> Which collection to add them to
-e <email> Existing user who is adding these items
-m <path> Create a log file for this import
-s <path> Location of the import files
Web-based submission

- Interface for the user:
  - Provide metadata
  - Upload the item
  - Review submission
  - Accept distribution license

- Limited configurability
  - Customize metadata

- DSpace / Manakin 1.5
Workflow

- Originating from
  - Web-based submission
  - Batch ingest
- Task pool
- 3-step process
- Separate review at each step
Workflow steps

Unfinished Submission -> Workflow Step 1 -> Workflow Step 2 -> Workflow Step 3 -> Available in DSpace

- Submit
- Accept
- Reject
- Edit Metadata

Accept
Accept
Accept

DSpace / Manakin Training

DSpace 1.5.1

TDL Training

May 13th, 2009

Scott Phillips
Texas A&M University
Texas Digital Library

Steven Williams
University of Texas @ Austin