8:30-9:45  Introductions
- Presenter
- Attendees
- Workshop Overview

Introduction to Manakin
- What is XMLUI?
- Interesting Manakin Sites
- Three Development Tiers
- The Road to DSpace 2.0
- Useful Resources

Technical and Conceptual Overview of DSpace and Manakin
- Anatomy of a DSpace Request
- Anatomy of a DSpace Request using Manakin

9:45-10:00  Break (15 minutes)

10:00-11:15  Technical and Conceptual Overview of DSpace and Manakin (continued)
  - Technical Details
    - Operating System
    - Database Server
    - Application Server
    - DSpace Live DVD
  - DSpace Basics
    - Configuring DSpace
    - Building & Deploying DSpace
    - Analyzing Log Files
    - Demonstrate DSpace Configuration Change

Introduction to Manakin's Style Tier
- Coding at the Style Tier
  - XHTML Basics
  - Review of CSS Selectors & Syntax
  - Manakin's Built-in Selectors
- Exploring with Firebug
  - Finding the Right Selectors
  - Demonstrate Design Changes with CSS

11:15-11:30  Break (15 minutes)

11:30-12:30  Hands-on with Manakin's Style Tier
- Configuring the Work Environment
  - Installing the oXygen XML Editor
Installing the Firebug Plug-in

Major Reference Theme Sections

Eight CSS Modifications
- Replace Manakin Logo
- Change Background & Border Colors
- Redesign Footer
- Move Menus to the Left
- Modify Border Style
- Change Text Colors
- Change Link Colors
- Change Bullet Colors

12:30-1:30 Lunch

1:30-2:30 **Introduction to Manakin's Theme Tier**
- Coding at the Theme Tier
- How Manakin Uses XML, XPATH, & XSL
  - Overriding Default Recipes
  - Accessing Relevant XSL Files
  - Accessing Generated XSL, DRI, & Mets Files
- Theme Cascading
- Why use an XML editor?
- Crash Course on XSL
  - What does XSL do?
  - Syntax and Use

2:30-2:45 Break (15 minutes)

2:45-4:00 **Introduction to Manakin's Theme Tier (continued)**
- Crash Course on XPATH
  - What does XPATH do?
  - Syntax and Use

**Hands-on with Manakin's Theme Tier**
- Configuring Our Environment
  - Starting a Custom XSL Stylesheet
  - Configuring Manakin to Use Custom XSL Stylesheets
- Five XSL/XML Customizations
  - Remove Main Page Search Recipe

4:00-4:15 Break (15 minutes)

4:15-5:30 **Hands-on with Manakin's Theme Tier (continued)**
- Modify Header Recipe
- Introduce New Footer Links
- Add Static Content
- Theme a Specific Collection

**Wrap-up**
- Very Brief Overview of Coding at the Aspect Tier
- Questions & Answers
Developing Interfaces and Interactivity for DSpace with Manakin

Part 1: Introduction to Manakin

Eric Luhrs
Digital Initiatives Librarian, Lafayette College
&
NITLE Technology Fellow for DSpace and Manakin

Presented at NITLE Information Services Camp
Smith College, June 3, 2009

With curricular and technical assistance from
Eric Jansson, NITLE
Overview of Part 1

• What is Manakin?
• Interesting Manakin sites
• Three development tiers
• The road to DSpace 2.0
• Useful resources
What is Manakin?

• Interface development framework
  – Toolkit for development of new interfaces and functionality
  – Coding at any of 3 levels to reflect organizational needs and capabilities
  – General-purpose, front-facing technology

• “Default” interface for DSpace as of 1.5
Why Use Manakin?

• Because Manakin is easier to work with and maintain, institutions will be more likely to customize:
  – Java experience unnecessary for development within first two tiers
  – Ability to share customizations and re-use with new versions of DSpace
  – Ability to create new interfaces and features for specific items, communities, and collections
In the Beginning, there was JSP

• “difficult and expensive to modify, and reinforces a cookie-cutter approach to the user interface” *
  - Changes made require exposure to Java (difficult)
  - New version of DSpace can require updates to past JSP work (expensive)
  - Changes made are repository wide (cookie cutter)

The Original DSpace Interface

MIT Libraries' Dome

Dome is home to a growing collection of digitized content of the MIT Libraries. Currently, Dome holds a collection of over 25,000 digitized images of Art and Architecture from the Rotch Visual Collections. New collections of photographs, maps, documents, and other materials will be added over time to enrich the Dome collection of online resources.

Communities in Dome

Select a community to browse its collections.

- MIT Communications Forum
- Perceptual Form of the City
- Rotch Visual Collections Online

All items in MIT Libraries' Dome are protected by original copyright, with all rights reserved, unless otherwise indicated.

http://dome.mit.edu/
What Can Manakin Do?

- Branding
- Repository
- Community
- Collection
- Item

http://libmedia.willamette.edu/xmlui/
What Can Manakin Do?

- Visualize Metadata

http://txspace.tamu.edu/handle/1969.1/2490
What Can Manakin Do?

- Create Complex Objects
- Re-design interfaces for better usability (e.g. gallery view here, with magnification)

http://dspace.lafayette.edu/handle/10385/93
What Can Manakin Do?

• Create new page layouts
• Add new components
• Incorporate static content

https://www.policyarchive.org/
What Does Manakin Enable the Community to Do?

Past versus Present

Photograph by Driek Heesakkers

Open Repositories 2009, Georgia Tech
http://www.flickr.com/photos/driek/3554761746/
Then
Then

Traditional DSpace

Flavor 1

Flavor 2

Flavor 3

Institution A

Institution B

Institution C

Now

DSpace

Flavor 1

Flavor 2

Flavor 3

Institution A

Institution B

Institution C

Now with Manakin
How Does Manakin Help?

• **Respond** to local needs
• **Modularity** that can be **shared**
  – More sustainable than JSP display & feature hacking
• Maintain code over time with reference to a **standard** (reduce and manage **risk**)
• Can potentially be implemented on top of other repositories
  – Manakin interfaces could be moved to other systems; **reuse** existing work
Manakin and Two Hot Topics: Integration

• The old distributed model: OAI-PMH
  – Hard to map
  – Difficult to maintain
  – Google retired support in April 2008 *

• A new distributed model: XML Sitemaps
  – Machine readable data format
  – New discovery format supported by Google.
  – Standard way to expose local resources
  – Manakin can generate Sitemaps automatically

* See http://googlewebmastercentral.blogspot.com/2008/04/retiring-support-for-oai-pmh-in.html
Manakin and Two Hot Topics: Cloud Computing

• Reorient IT from technology provision to services. So needs technologies which:
  – **separate** local integration from service management
  – allow **domain experts** access to configuration while **limiting** exposure to technical details
Three Levels of Manakin Development

Levels where you can code

1. Style Tier (render/display content)
2. Theme Tier (transform content)
3. Aspect Tier (generate content)
Coding at the Style Tier

• Capabilities: limited interface changes
  – Change style of elements: fonts, colors, borders, etc
  – Move interface elements around
    • Nesting is a constraint: no “un-nesting”
  – Introduce images into the layout (limited)
  – Hide (but not remove) interface elements
    • Not a security feature!

• Use cases: branding
Coding at the Theme Tier

• Capabilities
  - Introducing new interface elements
  - Removing and modifying existing interface components
  - Introduce new functionality (w/ limitations)

• Use cases: lots!
  - Seamless integration with website
  - Visualization of metadata
  - In-page item manipulation
  - Integration with other systems (i.e. producing machine-readable output)?
Coding at the Aspect Tier

• Capabilities
  – Introduce new functionality into DSpace

• Use cases
  – New workflows
  – Shopping cart
  – Etcetera
The Road to DSpace 2.0

- DSpace v1.5.2 released Spring 2009
  - New version of Cocoon
- DSpace v1.6 due Fall 2009
  - Stepping stone (major database changes)
- DSpace v2.0 due Winter 2010
  - New version of XMLUI (updated themes?)
  - New version of JSPUI provides theming
Resources

- DSpace Live CD
  - [http://cadair.aber.ac.uk/dspace/handle/2160/565](http://cadair.aber.ac.uk/dspace/handle/2160/565)
- DSpace wiki site
- Stuart Lewis’ Blog
  - [http://blog.stuartlewis.com](http://blog.stuartlewis.com)
- “Manakin: A New Face For DSpace”
  - [http://www.dlib.org/dlib/november07/phillips/11phillips.html](http://www.dlib.org/dlib/november07/phillips/11phillips.html)
- Theme Tutorial: Getting Started with Manakin Themes
  - [http://di.tamu.edu/projects/xmlui/resources/ThemeTutorial.pdf](http://di.tamu.edu/projects/xmlui/resources/ThemeTutorial.pdf)
- Recorded presentations from DSpace User Group Meeting at OR09 (good blog too)
End of Part 1
Developing Interfaces and Interactivity for DSpace with Manakin

Part 2: Technical and Conceptual Overview of Dspace and Manakin

Eric Luhrs
Digital Initiatives Librarian, Lafayette College
&
NITLE Technology Fellow for DSpace and Manakin

Presented at NITLE Information Services Camp
Smith College, June 3, 2009

With curricular and technical assistance from
Eric Jansson, NITLE
Overview of Part 2

1. Anatomy of a DSpace request
2. Anatomy of a DSpace request using Manakin
3. Technical details
4. Configuration details
5. The Maven build process
6. Re-configuring DSpace (demo)
Anatomy of a DSpace Request

What happens when a user submits a request to your DSpace server?
Anatomy of a DSpace Request

1. Internet
2. Application Server
3. DSpace
4. Database Server

- Operating System
- User Interface Layer
- Core Business Layer
- Data Access Layer

Browser
What Exactly Does Manakin Do?

• Converts XML-encoded metadata into XHTML, and other formats
• Applies static CSS to dynamic XHTML
• Provides mechanism to style segments of the repository in different ways
• Allows for the development of additional features called Aspects
Anatomy of a DSpace Request using Manakin (XMLUI)

Operating System

User Interface Layer

Core Business Layer

Data Access Layer

Browser

Internet

Application Server

Webapps

Database Server
**XMLUI Architecture**

1. **Generate content**
   - Aspects

2. **Transform content**
   - DRI Doc (XML)
   - Theme (XSL)

3. **Render/display content**
   - XHTML
   - CSS

Cocoon pipeline

Web Browser

**Interface**

XMLUI

**Business**

**Database**
The Technical Details

• Must you fully understand?
  - No, but....
  - Basic understanding will help diagnose problems
  - Many skill levels represented here
    • Some really NEED this knowledge (and may already have it)
    • Will help others UNDERSTAND their tech people
Components: The Operating System

• OS hosts the rest of the components
• DSpace runs on Linux, Unix, Mac, Windows
  – Cross-platform Java application
  – Development vs. production
• Linux and Unix used most
  – Stronger production environments
  – More documentation
  – Better community support
Components: The Database Server

- Database server holds:
  - User info
  - Item info
  - Collection info
- Database servers supported by DSpace:
  - Oracle
  - PostgreSQL
Components: The Application Server

- Also called a “servlet engine”
- Allows users to run compiled applications remotely through a web browser
- Application servers supported by DSpace:
  - Jetty
  - Caucho Resin
  - Apache Tomcat
  - Others too
The DSpace Live DVD: What is it?

• A read-only DSpace Server
• Changes saved in memory, not persistent
• Exploration with no risk
• Requires a bootable DVD drive
• OS = Ubuntu Linux
  – Database Server = PostgreSQL
  – Application Server = Apache Tomcat
    • DSpace = v1.5.1 beta
Tomcat: Config, Logs, Starting and Stopping

- Base directory
  `/etc/tomcat5.5/`

- Log file
  `/var/log/syslog`

- Starting and stopping
  `/etc/init.d/tomcat5.5 [start | stop | status | restart]`
DSpace: Config, Logs, Starting and Stopping

- Main configuration file
  `/opt/dspace/dspace/config/dspace.cfg`

- Log file
  `/opt/dspace/dspace/logs/dspace.log`

- Starting and stopping (same as Tomcat)
  `/etc/init.d/tomcat5.5 [start | stop | restart | status]`
Editing the DSpace Configuration File

• Choose your favorite text editor
  (GUI: Text Editor, CLI: nano, vi, etc)

**Terminal:** nano /opt/dspace/dspace/config/dspace.cfg
**Text Editor:** File → Open /opt/dspace/dspace/config/dspace.cfg

Various options represented by: `$variable = value$
  example: dspace.dir = /opt/dspace/dspace
The Maven Build Process

- **Make temporary changes in Install directory**
  - Install dir = /opt/dspace/dspace

- **Make permanent changes in Source directory**
  - Source dir = /opt/dspace/dspace-src

- **Rebuild Source directory**
  - Example: cd /opt/dspace/dspace-src
  mvn package

- **Deploy changes from Source dir to Install dir**
  - Example: cd /opt/dspace/dspace-src/dspace/target/dspace-1.5.1-SNAPSHOT-build.dir
  ant update
Analyzing Log Files

• Tomcat and DSpace report problems are either logged to the browser screen, or available in the following log files:

  /opt/dspace/dspace/log/dspace.log
  /var/log/syslog

• Some problems are obvious, others can be solved by searching or posting questions to mailing DSpace mailing lists
Demo: Re-configuring DSpace

- Modify DSpace configuration
- Restart Tomcat application server
- Verify changes on local Manakin site
End of Part 2
Part 3: Introduction to Manakin’s Style Tier

Eric Luhrs
Digital Initiatives Librarian, Lafayette College
&
NITLE Technology Fellow for DSpace and Manakin

Presented at NITLE Information Services Camp
Smith College, June 3, 2009

With curricular and technical assistance from
Eric Jansson, NITLE
Overview of Part 3

1. Coding at the style tier
2. Review of XHTML and CSS
3. Exploring with Firebug
Coding at the Style Tier

• Required Skills
  – CSS & XHTML

• What CAN’T be done with CSS?
  • Remove (not just hide) existing interface elements
  • Introduce new interface elements (such as menus, zoom features, etc)
  • Alter the positioning of elements in ways that violate nesting of HTML elements
Basic XHTML Requirements

- XHTML is HTML that is compatible with XML. Building off your HTML knowledge:
  - All tags must be closed. No more: `<hr>`; must be `<hr></hr>`
  - Tags with no content can look like this: `<hr/>
  - XML is case-sensitive. No more: `<p>paragraph text </p>`
  - Elements cannot overlap. No more: `<p>Paragraph <em>emphasized</em></p>`
  - Attributes must be quoted. No more: `<table border=3>`
How Does Manakin use CSS?

CSS: adds style information to HTML

XHTML

```html
<div>
    <h2>
        My Item
    </h2>
</div>
```

CSS

```css
h2 {
    font-weight:bold;
    font-size: 16pt;
    color: blue;
}
```

Browser Display

Image of a browser displaying stylized content.
Example CSS Rule

CSS Rules look like this:

```
p {  
  color: black;  
  font-size: 12pt;  
}  
```

This means: "Make all text between `<p>` (paragraph) tags black, 12pt font"
Assigning Style: XHTML Tags

There are 3 types of "selectors":
1. HTML tags
2. classes
3. ids

To return to our example:

```html
p {
    color: black;
    font-size: 12pt;
}
```

This is an example of applying style to an XHTML tag
Assigning Style: Classes

• The '.' denotes a class
• `.larger` is the class 'larger'
• A rule for the class 'larger' would look like this:

```css
.larger {
    font-size:24pt;
}
```

We would reference this class in HTML like this:

```html
<p class="larger">some text</p>
```
Assigning Style: IDs

- The `'#'` denotes an id
- `#main` is the id 'main'
- A rule for the id 'main' would look like this:

```css
#main {
    font-size:16pt;
    background-color:gray;
}
```

We would reference this class in HTML like this:

```html
<div id="main">some text</div>
```

NOTE: each ID should only be used once in each HTML page (it's an id after all: a unique identifier)
Combining Rules for XHTML Tags, Classes and IDs

You can combine rules, reading left to right:

```
  p.larger {...}
```

Means "apply this rule to all `<p>` (paragraphs) that are of the class 'larger'”

```
  #main p {...}
```

Means "apply this rule to all `<p>` (paragraphs) in the element denoted by ‘main’”

```
  #main p.larger {...}
```

Means "apply this rule to all `<p>` (paragraphs) that are of the class 'larger' in the element denoted by ‘main’”
Examples continued:

```
div#main p.larger span.larger  {...}
```

Means "apply this rule to all `<span>` elements of the class 'larger' that are in `<p>` (paragraphs) that are of the class 'larger' that are in the element denoted by "main.” As in:

```
<div id="main">
   <p class="larger">
      <span class="larger">some text</span>
   </p>
</div>
```
What do these mean?

```html
td {...}
ul.ds-artifact-list {...}
div#ds-user-box p {...}
div#ds-search-option input.ds-text-field {...}
form.ds-interactive-div li.last {...}
```
CSS References

There is much more to CSS:

– [http://www.w3schools.com/Css/default.asp](http://www.w3schools.com/Css/default.asp)
The “Style” of Manakin’s Default Themes

1. HTML lists are used structurally a lot; know how to:
   – Make them display horizontally
   – Remove indentation and add margins
   – Remove bullets or markers

2. Elements often support multiple classes; know how to:
   – Select the right style to modify
   – Firebug can help (demonstration)
Exploring Manakin with Firebug

- Using Firebug
  - Click “bug” in lower-right corner of Firefox
  - Click “inspect” then point to elements

- Note: Firebug is useful for exploring CSS rules and properties, but it cannot save changes you make (demo)
End of Part 3
Developing Interfaces and Interactivity for DSpace with Manakin

Part 4: Hands-on with Manakin’s Style Tier

Eric Luhrs
Digital Initiatives Librarian, Lafayette College
&
NITLE Technology Fellow for DSpace and Manakin

Presented at NITLE Information Services Camp
Smith College, June 3, 2009

With curricular and technical assistance from
Eric Jansson, NITLE
Overview of Part 4

1. Configuring our work environment
2. Major Reference theme sections
3. Eight hands-on CSS modifications
Configuring Our Work Environment

Step 1: Boot DSpace Live DVD

- Insert DSpace Live DVD into drive
- Boot (or reboot) computer
- Login with username/password “dspace”
  - Optional:
    - To change desktop background:
      - Right click on desktop, select “wallpaper”, and use color picker to choose a color, click “close”
    - To change terminal colors:
      - Double click “Terminal” icon, click Edit ➔ Current Profile, choose “Colors” tab, uncheck “use colors from system theme”, select “white on black” (or something else) from dropdown box, choose “Linux Console” from list of “Built-in schemes”
Configuring Our Work Environment
Step 2: Fix a Few Things

• Connect to the network by left clicking icon to left of speaker icon on upper-right taskbar, and choosing an available network

• Type the following command into a “Terminal” window (note space before trailing period):
  • sudo chown -R dspace:dspace .
Configuring Our Work Environment

Step 3: Install oXygen XML Editor

- Download and uncompress oXygen by typing the following commands into a “Terminal” window:
  - `wget http://mirror.oxygenxml.com/InstData/Editor/All/oxygen.tar.gz`
  - `tar xzvf oxygen.tar.gz`

- Create a desktop icon by right clicking on the desktop, choosing “Create Launcher” and providing the following values, before clicking “OK”:
  - Name: oXygen
  - Command: `/home/dspace/oxygen/oxygen.sh`
Configuring Our Work Environment

Step 4: Register the oXygen XML Editor

- Start the Firefox web browser by clicking the icon in the upper task bar
  - Bookmark http://tinyurl.com/manakin/
  - Click “license key” and copy all 9 lines

- Start the oXygen XML editor by double clicking the desktop icon
  - Paste all nine lines into “use license key” section of registration screen and click “okay”

- Close the oXygen editor for now
Configuring our work environment
Step 5: Install Firebug Plug-in

- Start the Firefox web browser by clicking the icon in the upper task bar
- Required: Install the Firefox Firebug plug-in:
  - Click "Add to Firefox"
  - Click "Install Now"
  - Click "Restart Firefox"
  - Close "Add-ons" window
  - Verify that Firebug icon is visible in the lower right corner of the Firefox status bar
- Return to http://localhost:8080/dspace-xmlui
Configuration Complete!

• NOTE: because we cannot save changes to the DVD, the previous steps will need to be repeated each time the Linux Live DVD is rebooted

• We are now ready to customize Manakin’s style tier
Eight Hands-on CSS Modifications

1. Replace Manakin logo
2. Change background & border colors
3. Redesign footer
4. Move menus to left
5. Modify border style
6. Change heading colors
7. Change link colors
8. Change bullet colors
1. Disable CSS in Firefox: View → Page Style → No Style

2. Re-enable CSS in Firefox: View → Page Styles → Basic Page Style

3. To determine which external style sheets are used to create this design, right click on the page and choose “View Page Source”. Look for the “link” line of type “text/css”. We can find this file at the following absolute system path: /opt/dspace/dspace/webapps/xmlui/themes/Reference/lib/style.css
Observe the major Reference them sections we will work with
1. Download NITLE logo from http://tinyurl.com/manakin and save to 
   /opt/dspace/dspace/webapps/xmlui/themes/Reference/images/nitle.png
2. Start the oXygen editor and open the CSS file:
   /opt/dspace/dspace/webapps/xmlui/themes/Reference/lib/style.css
3. Close “Project” pane
4. Return to Firefox, enable Firebug’s “Inspect” feature, and point to the Manakin
   logo
5. Return to oXygen and sort the “Outline” pane by “Selector” and locate the
   div#ds-header a span#ds-header-logo rule
6. Change filename specified in div#ds-header a span#ds-header-logo rule to
   background-image: url(“../images/nitle.png”) from manakin_logo.jpg
7. Save file in oXygen and reload Firefox to see new logo
8. Explore with Firebug to determine which properties control logo spacing
9. Use oXygen to change height property of div#ds-header a span#ds-header-
   logo to height: 100px from 80px
10. Save file in oXygen and reload in Firefox to see new logo spacing
11. Use oXygen to change height property of ul#ds-trail rule to margin-top: 100px
    from 80px
12. Save file in oXygen and reload in Firefox to see new logo spacing
1. This step is tricky because Manakin's Reference theme uses RGB color codes in some places, and the more standard Hexadecimal codes in others.

2. Use Firebug to determine color values used for original background (tan: #FFFFF0) and border (beige: #F0F0D2) colors on right-hand ds-options menu, which will be the same colors used elsewhere in the theme.

3. Use the oXygen editor to locate the RGB values listed div.ds-option-set rule (not div#ds-option-set).

4. Start oXygen’s “Find/Replace” feature: Find → Find/Replace or CTRL-F.

5. We want to replace the RGB values with Hexadecimal values for light blue background color (#9CADBF) and a dark blue border color (#3D536B).

6. Use oXygen editor to find all occurrences of rgb(255, 255, 240) and “Replace All” with new background color #9CADBF.

7. Use oXygen editor to find all occurrences of rgb(240, 240, 210) and “Replace All” with new border color #3D536B.

8. Save file in oXygen and reload Firefox.
1. Use Firebug to determine which rules control footer logo and text so we can hide them
2. Locate these rules using the oXygen editor
3. Hide footer logo and text by adding new `visibility: hidden` property to `span#ds-footer-logo` and `div#ds-footer p` rules
4. Save in oXygen and reload Firefox to verify result
5. Use Firebug to determine which rule control menu height
6. Decrease footer height by changing `div#ds-footer` property to `height: 25px` from 80px
7. Save in oXygen and reload Firefox to verify result
8. To make the footer match our color scheme, also add a new `background-color: #9CADBF` property to `div#ds-footer` rule
9. Save in oXygen and reload Firefox to verify result
10. Use Firebug to determine which rule controls the position of footer links
11. Reposition footer links by changing `div#ds-footer-links` property to `top: 5px` from 57px
12. Save in oXygen and reload Firefox to verify result
1. Use Firebug to determine which rules control the menu and body

2. Locate these rules using the oXygen editor

3. Reverse menu and body positions by switching `div#ds-body` property to `float:right` and `div#ds-options` property to `float:left`

4. Save in oXygen and reload Firefox to verify result
1. Use Firebug to determine which rules controls the ds-body border width

2. Use oXygen editor to locate div#ds-body rule

3. Start oXygen’s “Find/Replace” feature: Find → Find/Replace

4. Use oXygen editor to find 2px solid #3D536B and replace with 1px solid #3D536B

5. Save in oXygen and reload Firefox to verify result

6. Use oXygen editor to find 3px solid #3D536B and replace with 1px solid #3D536B

7. Save in oXygen and reload Firefox to verify result

8. To darken top border of breadcrumb trail, locate ul#ds-trail rule in oXygen editor and change property to border-top: 1px solid #3D536B from 2px solid rgb(245, 245, 216)

9. Save in oXygen and reload Firefox to verify result
1. Use Firebug to determine which rules make text orange

2. Using the oXygen editor, locate `div#ds-body h1` and `div#ds-body-h2` (even though we don’t have this element here, we don’t want orange text popping up elsewhere) and change properties of both to `color: #3D536B` from `#DF6E00`

3. Save in oXygen and reload Firefox to verify result

4. Using the oXygen editor, locate `div#ds-options h3` and `div#ds-options h4` and change properties of both to `color: #3D536B` and `background-color: #3D536B`

5. Save in oXygen and reload Firefox to verify result
1. Use Firebug to determine which rule controls breadcrumb links
2. Locate the rule using the oXygen editor
3. Add new property `color: #FFFFFF` to `div#ds-header a` rule
4. Save in oXygen and reload Firefox to verify result
5. We can also add a hover rule add an underline when someone points to the ds-header link. We do this by creating a new rule called `div#ds-header a:hover`, which has the same `color: #FFFFFF` property, but also adds the `decoration:underline` property.
6. Save in oXygen and reload Firefox to verify result
7. Now we want to expand both of these rules to include menu advanced search, menu, and footer links. Start by using Firebug to determine which rule controls the advanced search link.
8. Expand the link rule to `div#ds-header a, div#ds-search-option a` (note that this is now one rule with two selectors), and the link hover rule to `div#ds-header a:hover, div#ds-search-option a:hover`
9. Save in oXygen and reload Firefox to verify result
10. When analyzing the menu links in Firebug, you will notice that there are no rules that control them. That’s okay, because we can create our own by adding `div#ds-options a` and `div#ds-options a:hover` selectors to our two rules.
11. Save in oXygen and reload Firefox to verify result
12. The same is true for `div#ds-footer-links`, but we can create these too by adding our own `div#ds-footer-links a` and `div#ds-footer-links a:hover` to our two rules.
13. Save in oXygen and reload Firefox to verify result
1. Use Firebug to determine which rule controls menu bullets

2. Locate the rule using the oXygen editor

3. Change bullet color property in `div#ds-options li` to `color: #3D536B` from `rgb(100, 100, 50)`

4. Save in oXygen and reload Firefox to verify result
Admire your work!
End of Part 4
Developing Interfaces and Interactivity for DSpace with Manakin

Part 5: Introduction to Manakin’s Theme Tier

Eric Luhrs
Digital Initiatives Librarian, Lafayette College
&
NITLE Technology Fellow for DSpace and Manakin

Presented at NITLE Information Services Camp
Smith College, June 3, 2009

With curricular and technical assistance from
Eric Jansson, NITLE
Overview of Part 5

• Coding at the theme tier
• How Manakin uses XML, XSL, & XPATH
• Why use an XML editor?
• Crash course on XSL
• Crash course on XPATH
Coding at the Theme Tier

• Required Skills
  – CSS & XHTML
  – XML editing
  – Some knowledge of XSL & XPATH
  – [plus development environment]
Don’t Be Misled

• XSL is not a simple language
• For that matter, neither are CSS and XHTML
• Programming XSL is different than many kinds of programming (rule-based rather than procedural or object oriented)
XSL/ XML: A Growing Area

• Common language of configuration and integration
  – Drupal
  – Sakai OSP
  – Manakin

• Goal is to model local domain knowledge (or structures) by abstracting it into a machine readable format

• Wizard applications will help reduce need to know XML/XSL (and others), but how much?
How Manakin uses XML, XPATH, XSL
Exploded View of Relevant XSL Files

Manakin → XML → i18n → XSLT → Output

DIM-Handler.xsl
General-Handler.xsl
MODS-Handler.xsl
QDC-Handler.xsl
structural.xsl

dri2xhtml.xsl

[theme].xsl
Default Manakin XSL Recipe

`<Recipe1>
  <Content>
    AAA
  </Content>
</Recipe1>`

Manakin → XML → i18n → XSLT → `<p>
  AAA
</p>`

XSL files:
- `dri2xhtml.xsl`
- `structural.xsl`
- `[theme].xsl`
Overriding Default Manakin XSL Recipe
Accessing Relevant XSL Files

In your theme directory, for instance:
/opt/dspace/dspace/webapps/xmlui/themes/Reference/Reference.xsl

We only need the following core Manakin XSL file:
/opt/dspace/dspace/webapps/xmlui/themes/dri2xhtml/structural.xsl

Aggregates five core Manakin XSL files for use by themes

In your theme directory, for instance:
/opt/dspace/dspace/webapps/xmlui/themes/Reference/Reference.xsl
Structure of Source XML & DRI Documents

• Looks complicated, but only three parts:
  – Body: what is the interface content?
    • Structured after Text Encoding Initiative (TEI)
  – Options: what can I do or where can I go from here?
  – Metadata (administrative/technical)
    • What do I know about the user (e.g. logged in?)
    • What is the context?
      – What do I know about this page?
      – What do I know about this repository?

• Theme developer can:
  – Use or ignore any of these
  – Setup logic based on these
Accessing Generated XML, DRI, METS Files

- Easy to access XML data at any stage of conversion
  - Default XHTML browser output
  - Native XML: before internationalization
  - DRI (Digital Repository Interface) XML: after internationalization
  - METS (Metadata Exchange & Transmission Standard) XML
Theme Cascading

- Repository
  - Community
    - Collection
      - Item
    - Community
      - Sub-Community
        - Collection
        - Item
        - Collection
        - Item
Theme Cascading

- Repository
  - Community
    - Collection
      - Item
      - Custom theme
  - Community
    - Sub-Community
      - Collection
        - Collection
          - Item
          - Item
      - Custom theme
Why Use an XML Editor?

• **XML is fussy**
  - Tags are case sensitive
  - Tags must be closed
  - Tag attributes must be quoted
  - Tags cannot overlap
  - Requires character codes: `&lt;` (<), `&gt;` (>, `&amp;` (&), `&quot;` ("), `&apos;` (‘)

• **Key benefits**
  - Prevent all of the above
  - Syntax highlighting
  - Validate XML with custom Schemas
  - Debug XSLT transformation
  - Build XPATH expressions
Crash Course on XSL: What Does it Do?

XSL: transforms one XML file into another

XML:

```
<item>
  <title>
    My Item
  </title>
</item>
```

XSL:

```
<xsl:template match="item">
  <div>
    <xsl:apply-templates />
  </div>
</xsl:template>

<xsl:template match="title">
  <h2>
    <xsl:apply-templates />
  </h2>
</xsl:template>
```

XHTML:

```
<div>
  <h2>
    My Item
  </h2>
</div>
```
XSL elements allow us to apply processing logic to XML data, for example (from structural.xsl)

```xml
<xsl:template match="dri:body">
  <div id="ds-body">
      <div id="ds-system-wide-alert">
        <p>
        </p>
      </div>
    </xsl:if>
    <xsl:apply-templates/>
  </div>
</xsl:template>
```
The oXygen XML Editor allows you to associate XSL stylesheets with XML documents, and step through transformations (demo).
Crash Course on XPATH: What does it do?

XPATH matches nodes in XML documents

XML Data

```xml
<page>
  <item>
    <title>My Item</title>
    <author>Somebody</author>
    <author>Someone else</author>
  </item>
</page>
```

Sample XPATH Syntax:

```
/  
/page  
/page/item  
/page/item/author  
item  
/page/item/author[1]  
//author  
//item
```
Crash Course on XPATH: Demo

• The oXygen XML Editor provides tools that make it easier to build complicated XPATH expressions (demo)
XML, XSL, & XPATH Resources

• A few links to get you started:
  - http://zvon.org
  - http://www.w3.org/XML/
  - http://www.w3schools.com/xml/
  - http://www.w3.org/Style/XSL/
  - http://www.w3schools.com/xsl/
  - http://www.w3.org/TR/xpath
  - http://www.w3schools.com/XPath/
End of Part 5
Developing Interfaces and Interactivity for DSpace with Manakin

Part 6: Hands-on with Manakin’s Theme Tier

Eric Luhrs
Digital Initiatives Librarian, Lafayette College
&
NITLE Technology Fellow for DSpace and Manakin

Presented at NITLE Information Services Camp
Smith College, June 3, 2009

With curricular and technical assistance from
Eric Jansson, NITLE
Overview of Part 6

1. Configuring our work environment
2. Five hands-on XML & XSL modifications
Configuring Our Work Environment
Step 1: Only If You Rebooted

- If you rebooted since the last hands-on section, you will need to repeat the steps listed on slides 4-7 of “Part 4: Hand-on with Manakin’s Style Tier.” These will include:

  - Connecting to network
  - Fixing a few things
  - Installing oXygen
  - Registering oXygen
  - Installing Firebug
Configuring Our Work Environment
Step 2: Verify That You Have New Logo

• If not, download from http://tinyurl.com/manakin
  – right click NITLE logo and choose “Save Image As”
  – save after navigating to the following directory
    /opt/dspace/dspace/webapps/xmlui/themes/Reference/images
  – replace file if prompted to do so
Configuring Our Work Environment
Step 2: Standardize CSS Configuration

- From [http://tinyurl.com/manakin](http://tinyurl.com/manakin) click the "modified style.css file" link.
  - Click `File → Save Page As` to save a local copy here: /opt/dspace/dspace/webapps/xmlui/themes/Reference/lib
  - Replace the file if prompted to do so
  - Right click the file from Firefox's "Downloads" window and select "Open containing folder"
  - Right click style.css in the file browser, select "Properties" then choose the "Permissions" tab. Verify that "Access" is set to "Read and write", then click "Close"
Configuring Our Work Environment

Step 3: Copy XSL Stylesheet to Theme Dr

- Use File Browser (Places → Computer → File System) to copy existing XSL stylesheet

  /opt/dspace/dspace/webapps/xalui/theme/Classic/Classic.xsl

  and paste to this directory

  /opt/dspace/dspace/webapps/xalui/themes/Reference

  then right-click to rename file

  Reference.xsl
Configuring Our Work Environment

Step 4: Tell Theme to Use XSL Stylesheet

• Open Reference theme’s sitemap.xmap file in oXygen (as XML document):

```
/opt/dspace/dspace/webapps/xalui/theme/Reference/sitemap.xmap
```

find the following lines

```
<map:transform src="/...dri2xhtml.xsl"/>
```

and **save** after changing the second one to

```
<map:transform src="/Reference.xsl"/>
```

(another link in the chain: Reference.xsl will import dri2xhtml.xsl)
Configuring Our Work Environment
Step 5: Create Icon to Restart Tomcat

- Right-click the desktop and choose “Create Launcher” then provide the following values before clicking “OK”:

  Type: Application in Terminal
  Name: Restart Tomcat
  Command: `sudo /etc/init.d/tomcat5.5 restart`
Configuration Complete!

- NOTE: because we cannot save changes to the DVD, the previous steps will need to be repeated each time the Linux Live DVD is rebooted

- We are now ready to customize Manakin’s theme tier
Reference Theme with Previous CSS Changes

![Image of DSpace/Manakin Repository]

- **DSpace/Manakin Repository**
  - Welcome to the new Manakin interface in the DSpace digital repository. DSpace is a digital repository that acquires, preserves, and distributes digital materials. Keywords are important tools for increasing the repository's visibility, making it easier for researchers and others to discover and access information.

- **Search DSpace**
  - Enter a search in the box below to search DSpace.

- **Communities in DSpace**
  - Select a community to browse its collections.
1. Look at [http://localhost:8080/](http://localhost:8080/) and then get to raw XML by adding “?XML” to the end of the URL

2. Look for references to “front-page-search”

3. Open Reference.xsl in oXygen and add the following recipe just before the `</xsl:stylesheet>` line:

   ```xml
   <xsl:template match="dri:div[@n='front-page-search']">&nbsp;</xsl:template>
   ```

4. Save in oXygen, then double click “Restart Tomcat” icon and wait for Terminal window to close

5. Verify that main page search has been removed

6. Use firebug to determine new height and then use oXygen to add `min-height: 318px` property to `div#ds-body` rule in style.css
1. Visit [http://localhost:8080/dspace-xmlui](http://localhost:8080/dspace-xmlui) to demonstrate logo link

2. Use Firebug to find that the `<span>`'s id=ds-header-logo

3. Use oXygen to open and search for “logo” in /opt/dspace/dspace/webapps/xmlui/themes/dri2xhtml/structural.xsl

4. Analyze construction of `<a>` link

5. Copy buildHeader code and paste into Reference.xsl

6. Remove `<a>` link, leaving only the `<span>` element

7. Save in oXygen and double click “Restart Tomcat” icon and wait for Terminal window to close

8. Visit [http://localhost:8080/dspace-xmlui](http://localhost:8080/dspace-xmlui) to observe missing logo

9. Edit style.css to remove “a” requirement from logo rule, `div#ds-header a span#ds-header-logo`

10. Visit [http://localhost:8080/dspace-xmlui](http://localhost:8080/dspace-xmlui) to show logo without link
1. Use oXygen to search for and remove all `visibility: hidden` properties from footer rules in `style.css`

2. Reload browser and find a string to grab

3. Using oXygen, search for “Texas” in `/opt/dspace/dspace/webapps/xmlui/themes/dri2xhtml/structural.xsl`

4. Notice that “buildFooter” is the recipe that needs to be overwritten

5. Add following modified recipe to Reference.xsl

   ```xml
   <xsl:template name="buildFooter">
     <div id="ds-footer">
       <div id="ds-footer-links">
         <p>
           <a href="http://www.nitle.org">NITLE</a> | 
           <a href="http://www.smith.edu">Smith College</a>
         </p>
       </div>
     </div>
   </xsl:template>
   ```

6. Save in oXygen, then double click “Restart Tomcat” icon and wait for Terminal window to close

7. Verify that new links exist in footer
1. Open the file browser (Places → Computer → File System) and navigate to the Reference theme directory located at /opt/dspace/dspace/webapps/xmlui/themes/Reference
2. Right click to create a new folder called “static”
3. Navigate to the “static” folder
4. Right click to create two documents named “faq.html”
5. Visit http://localhost:8080/dspace-xmlui/static/faq.html to see that it is not readable
6. Open the Reference theme’s sitemap.xmap file with oXygen and search for the second occurrence of “themes/*/**”
7. Add the following map statements under the existing one
   <map:match pattern="*.html">
     <map:read src="static/{1}.html" mime-type="text/html" />  
   </map:match>
8. Save in oXygen, then double click “Restart Tomcat” icon and wait for Terminal window to close
9. Reload Firefox to verify that link exists
10. Perform steps from previous slide to add new ds-footer-link to new static content in Reference.xsl (note: do not include “static” directory in the URL)
11. Click new “FAQ” link to see static file
12. Add as many static pages as you like
13. Creating a "*.txt" version of this map statement will allow you to serve a robots.txt file, which is something that is currently broken with DSpace v1.5.1
1. Use the file browser (Places → Computer → Filesystem) to change directories to /opt/dspace/dspace/bin

2. Double click “create-administrator” select “Run in Terminal”, then provide email, first name, last name, and password.


4. Click “Communities & Collection” then “Create Community”

5. Visit new collection to show existing theme

6. Use oXygen to open /opt/dspace/dspace/conf/xmui.xconf (as XML document) and add the following line ABOVE the default theme declaration:

   `<theme name="Kubrick" handle="123456789/1" path="Kubrick/" />`

7. Double click “Restart Tomcat” icon and wait for Terminal window to close

8. Visit new collection to show Kubrick theme
Admire your work, again!
End of Part 6