Introducing Manakin: Overview & Architecture

Scott Phillips, Cody Green, Alexey Maslov, Adam Mikeal, and John Leggett
Manakin Overview
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
Look & Feel

▶ Style

▶ Integration
Branding

- Repository
- Communities
- Collections
- Items
Visualization

- Interpret metadata
- Link metadata
- Explain metadata
Modularity

- Extend
- Adapt
- Customize
Manakin vs JSP UI

JSP UI
- Difficult & expensive to extend
- One monolithic interface for all collections regardless of content

Manakin
- Modular design makes extensions easier
- Multiple interfaces that can adapt to the needs of a collection
- Metadata in native formats
It is ready now

- Manakin 1.0 released
  - Browsing, Searching
  - Authentication
  - Submission Workflow
  - Administration

http://di.tamu.edu/projects/manakin/
How do I get started?

- You will need an installation of DSpace
- First, install Manakin on your DSpace
- Next, design your new interface
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
What resources are available?

- Documentation
  - Aspect Developers Guide
  - Theme Writing Tutorial
  - Schema Reference Manual

- Mailing lists
  - dspace-dev
  - dspace-manakin

- JCDL 2007
  - DSpace Tutorial
Architecture
Apache Cocoon

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

Sitemap

sitemap defines the pipeline

Generator
Transformer X
Transformer Y
Serializer

SAX

<doc>
   <z/>
   <x/>
</doc>
<doc>
   <z/>
</doc>
<doc>
   <x/>
   <y/>
</doc>
<doc>
</doc>

Cocoon Pipeline
Three Components

- Aspects
- Themes
- DRI Schema
**Structural Overview**

**Content Generation**
- Aspect 1
- Aspect 2
- Aspect 3

**Style Application**
- Theme A
- Theme B
- Theme C

DI 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
- “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

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

Content Generation

Artifact
Browser
Eperson
Submission

Style Application
Theme B

<html>
<head> ...
</head>
<body>
<div>
<h1> viewing an item </h1>
<table>
 ... The item's metadata ...
</table>
</div>
<li><a href="/logout"> Logout </li>
<li><a href="/profile"> Profile </li>
<li><a href="/submissions"> Submissions </li>
</div>
</body>
</html>
Questions