What is the Problem?
A Tale of being too successful!

**DSpace 1.x:**
- Rapidly developed by HP/MIT developers.
- Gains immediate Popularity
- HP calls it a SUCCESS
- Original Team moves on to new projects.
- Adopters utilize local staff for customization.
- Small Team of overtaxed volunteer commiters is left.
- HP/MIT create Foundation and hand over IP.
What is the Problem?
A Tale of being too successful!

DSpace 1.x:
- Rapidly developed by HP/MIT developers.
- Gains immediate Popularity
- HP calls it a SUCCESS
- Original Team moves on to new projects.
- Adopters utilize local staff for customization.
- Small Team of overtaxed volunteer commiters is left.
- HP/MIT create Foundation and hand over IP.
What is the Problem?
The struggle of customization

- Monolithic
- No Interface Contracts
- Users Alter Implementation
- Alterations Conflict
- Contributions Stagnate
What is the Problem?
The struggle of contribution

- Code improvements often bound to research cycles
- Large projects take longer to complete
- DSpace branches evolve in parallel
- Conflicts + Dramatic Changes = Increased Risk
What is the Problem?
The struggle of contribution

- Code improvements often bound to research cycles
- Large projects take longer to complete
- DSpace branches evolve in parallel
- Conflicts + Dramatic Changes = Increased Risk

⇒ Result: Traffic Jam
Better Practices:
Awareness is half the battle.

- We have been very “Aware” of these problems!
- Community has already provided some solutions.
  - Plugin Manager, Event Manager, Usage Event.
  - Manakin XMLUI, Maven Adopted.
- Community knows there is more to do (DSpace 2.0).
  - Service Architecture, New Data Model, API Contracts
Better Practices:
Open Developer Community

COMMUNITY

- Developers
- Design Guidelines
- Modular Projects
- Common Workspace
Better Practices:
More Developers

- New Bylaws = New Developers
- More Eyes Keeping Code Safe
- Less “Gate-Keeping”
- Migrated SVN = New Commons
- New Neighbors = Cross-pollination
Better Practices: Better API Contracts for Modules

User

Code

Developer

Okay.... Here's how it works...
Better Practices: Modular Design Guides

Modules, Encapsulation and API Contracts

- Separate Functional Domains.
- Protect against tangled inter-dependencies.
- Reliable Integration Points
- Open Doors to Innovation.
Better Tools: What is Cocoon 2.2?

- A Modular Framework
- UI Service Contracts
- XML Pipelines and More

Cocoon Blocks
- Template
- FOP
- Forms
- etc.

Cocoon Core
- Sitemap Engine
- Pipelines
- Store
- etc.

Cocoon Spring Configurator
Cocoon Servlet Service Framework

Spring Framework
At first plugin architecture was needed to support modularity.

Blocks designed by Stefano Mazzocchi et al. 4+ years ago.

Compile time blocks before 2.2 no external contracts, not dynamic configurability.

Cocoon 2.2 implementation is Spring / Maven based with external contracts.

Leads to more modular and reconfigurable modules.

See more: http://wiki.apache.org/cocoon/ButterflyManifesto
What is a Cocoon Block?

- A Webapp Fragment.
- Servlets, Resources, Components, etc.
- External Contracts,
  - Inter Dependency
  - Polymorphism
- Configurable at Deployment.
Cocoon 2.2 Block
Based on Servlet API

- Block = Embedded Servlet Container
- Block properties = Servlet init-params
- Block Connections = Named Dispatchers
- Not Cocoon Specific.
- Usable with any Servlets you create.
Maven Projects

- Archetypes: Quickly generate DSpace Addon Projects
- Provide some of the basic wiring for Adding modules to DSpace
Easy Startup:
Maven DSpace Archetypes

foo$ mvn archetype:generate -DarchetypeCatalog=http://scm.dspace.org/catalog
[INFO] Scanning for projects...
[INFO] Searching repository for plugin with prefix: 'archetype'.
[INFO] -----------------------------------------------
[INFO] Building Maven Default Project
[INFO] task-segment: [archetype:generate] ( aggregator-style )
[INFO] -----------------------------------------------
...
Choose archetype:
l: http://scm.dspace.org/catalog -> dspace-archetype-block (Creates a DSpace 1.5.2 block)
Choose a number:  (l): 1
Define value for groupId:  org.dspace.xmli
Define value for artifactId:  xmlui-rdf
Define value for version:  1.0-SNAPSHOT:  2.0.0-SNAPSHOT
Define value for package:  org.dspace.xmli:  org.dspace.xmli.app.rdf
...
[INFO] BUILD SUCCESSFUL
[INFO] -----------------------------------------------
[INFO] Total time: 4 minutes 12 seconds
[INFO] Finished at: Tue May 19 23:49:12 EDT 2009
[INFO] Final Memory: 8M/15M
[INFO] -----------------------------------------------
Generated XMLUI Block

XMLUI Block structure:

- xmlui-rdf
  - src/main/java
    - org.dspace.xmlui.app.rdf
      - AdvancedFormTest.java
      - BasicFormTest.java
      - HTMLTest.java
      - InlineFormTest.java
      - Navigation.java
      - StructureTest.java
  - src/main/resources
    - COB-INF
      - sitemap.xmap
    - META-INF
      - cocoon
        - spring
          - block-servlet-service.xml
    - org
      - dspace
        - xmlui
          - app
            - rdf
              - sitemap.xmap
  - src
    - pom.xml
    - rcl.properties
Generated XMLUI Block

- Generated Code
- Cocoon Block Sitemap
Generated XMLUI Block

Generated Code
Cocoon Block Sitemap
Spring Services Config
Generated XMLUI Block

- Generated Code
- Cocoon Block Sitemap
- Spring Services Config
- XMLUI Aspect Sitemap
Generated XMLUI Block

- Generated Code
- Cocoon Block Sitemap
- Spring Services Config
- XMLUI Aspect Sitemap
- Maven Project
Demonstration
Cocoon Wiring

- Reduces uncertainty in overlaying
- Blocks secure address
- Modularity no longer hindered by monolithic files

Reference Theme

METS Block

My Theme

Extends

Uses

/xmlui/themes/reference

/xmlui/themes/mine

super: mount:
Cocoon Wiring

- Frees Up Address Space
- Modularity no longer hindered by monolithic files
  - web.xml
  - sitemap.xmap
Better Tools: DSpace in Cocoon 2.2

DSpace Typed Presentation Blocks
- Default
- Journal
- Book
- Custom

DSpace Service and Aspect Blocks
- DCMI
- Entity
- DRI
- RDF

Description Set
- Driven
- Presentation
- Content Model

Description Set
- Drive
- Content Models
DSpace 2.0 Initiative Partners

Mark Diggory
mdiggory@atmire.com