Person Attributes, PAGS, and DLM:

Considerations for Upgrading uPortal

Andrew Wills

JA-SIG Winter Conference, December 5th, 2006

Atlanta, GA

UNICON™

Three Questions

- Is there anyone already running a portal based on uPortal 2.5 using DLM?
- Is there anyone thinking about upgrading an existing portal installation to uPortal 2.5 in the next 6 months?
- Is there anyone thinking about implementing a brand new portal solution based on uPortal?

About Me & UNICON PS

- Member of UNICON's Professional Services team for about 18 months
- Worked with Cal Poly to upgrade their portal to v. 2.5 & DLM from v. 2.1 & SLM
- Currently working with University of Colorado Systems to upgrade to v. 2.5 & DLM as well
- Participated in a panel discussion at the June conference discussing the upgrading experiences from different institutions (6 of 7 moved to 2.5.x)

CAL POLY

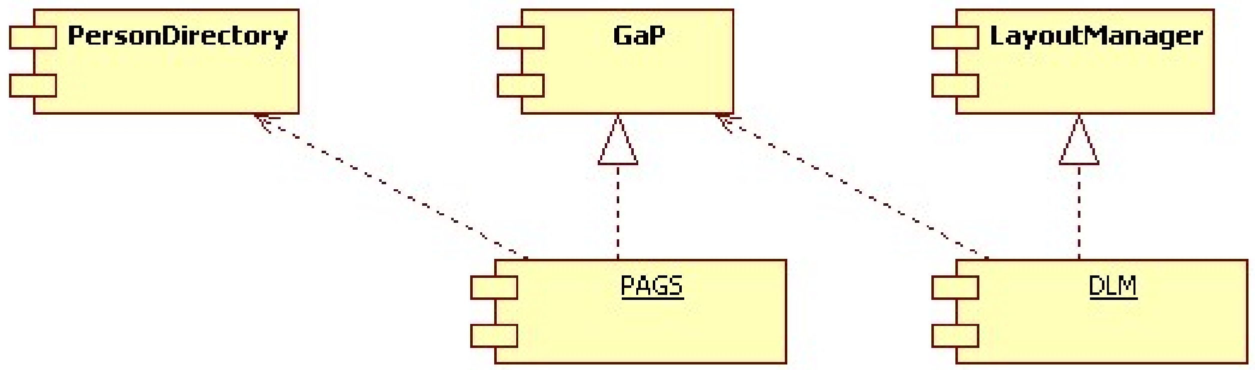
University of Colorado System

1. Subsystem Collaboration Overview
2. Person Attributes
3. PAGS
4. DLM
5. Popular & Compelling Paradigm

Subsystem Collaboration Overview

Subsystem Collaboration Overview

PersonDirectory, PAGS, & DLM



The combination of these technologies is greater than the sum of its parts

Collaboration Breakdown

- PersonDirectory is a subsystem, but PAGS & DLM are realizations of subsystems
- In other words...
  - There's more to GaP than PAGS, and DLM can leverage other realizations of GaP
  - DLM isn't the only option when it comes to Layout Management
- Together, these three technologies make up a popular and compelling paradigm for managing content and permissions in the portal

Collaboration Breakdown (Cont.)

- To put it another way...
  - DLM leverages the GaP subsystem, which can be driven by Person Attributes (through PAGS)
  - The things that are innovative about each of these technologies become more interesting when they are considered together

PersonDirectory

Three Flavors

- Person Attribute features have been expressed in three evolutions or flavors:
  - uPortal PersonDirectory service (uPortal 2.4 and earlier)
  - IPersonAttributeDao (uPortal 2.5)
  - Stand-alone PersonDirectory JA-SIG subproject (uPortal 2.6, 3.0)
- The second evolution is nearly identical to the third and is meant as a transition toward it

Evolution One: PersonDirectory Service

- Versions: uPortal 2.4.x & previous
- Static Entry Point(s):
  - org.jasig.portal.services.PersonDirectory.getUserDirectoryInformation() method
- Configuration: properties/personDirs.xml

Evolution One: PersonDirectory Service (Cont.)

- Supports the following data source types for person attributes
  - LDAP
  - RDBMS
- All necessary intelligence for working with both types of attribute sources is coded directly into PersonDirectory.java
- There is no facility for specifying alternate approaches at deployment time
- Data sources must be keyed by **username**

Evolution Two: IPersonAttributeDao

- Versions: uPortal 2.5.x
- Static Entry Point(s):
  - org.jasig.portal.services.PersonDirectory.getUserDirectoryInformation() method (backwards compatible)
  - org.jasig.portal.services.PersonDirectory.getPersonAttributeDao() method
  - org.jasig.portal.spring.PortalApplicationContextFacade.getPortalApplicationContext() method
- Configuration: properties/personDirectory.xml
- Data sources may key their queries from attribute that is already established

Evolution Two: IPersonAttributeDao (Cont.)

- Provides the following concrete IPersonAttributeDao implementations
  - LegacyPersonAttributeDao
  - CachingPersonAttributeDaoImpl
  - CascadingPersonAttributeDao
  - EchoPersonAttributeDaoImpl
  - JdbcPersonAttributeDaoImpl
  - LdapPersonAttributeDaoImpl
  - MergingPersonAttributeDaoImpl
  - StubPersonAttributeDao
- Additional, custom implementations of IPersonAttributeDao may be introduced through spring dependency injection

Evolution Three: PersonDirectory Subproject

- Versions: uPortal 2.6.x, 3.0
- Static Entry Point(s) [uPortal 2.6 only]:
  - [Same as Evolution Two]
- Configuration:
  - properties/personDirectory.xml (2.6); or
  - webapp/beans/portal/person-directory.xml (3.0)
- Data sources may key their queries from attribute that is already established

Evolution Three: PersonDirectory Subproject (Cont.)

- Provides the following concrete IPersonAttributeDao implementations
  - CachingPersonAttributeDaoImpl
  - CascadingPersonAttributeDao
  - EchoPersonAttributeDaoImpl
  - MergingPersonAttributeDaoImpl
  - RegexGatewayPersonAttributeDao
  - StubPersonAttributeDao
  - MultiRowJdbcPersonAttributeDao
  - SingleRowJdbcPersonAttributeDao
  - LdapPersonAttributeDaoImpl
  - DeclaredRulePersonAttributeDao
- Additional, custom implementations of IPersonAttributeDao may be introduced through spring dependency injection

Using Evolution Three in uPortal 2.5

- Take the following steps to use the Stand-alone JA-SIG PersonDirectory subproject in uPortal 2.5.x
  1. Remove files under source/org/jasig/portal/services/persondir (4 packages)
  2. Add person-directory-1.0.0-RC2.jar, which may be obtained by building the PersonDirectory subproject w/ Maven 1.x
  3. Change build.xml to add the person-directory jar to the compilation classpath and build output
  4. Change configuration appropriately in properties/personDirectory.xml

Custom IPersonAttributeDao Implementations

- You can create your own IPersonAttributeDao implementations to accomplish things not already contemplated by the PersonDirectory subproject
- IPersonAttributeDao defines three methods:
  - Map getUserAttributes(Map seed)
  - Map getUserAttributes(String uid)
  - Set getPossibleUserAttributeNames()
- Extend AbstractDefaultAttributePersonAttributeDao to bring that number down to two

Custom IPersonAttributeDao Implementations (Cont.)

- Here are examples of custom IPersonAttributeDao implementations you could create:
  - Doppelgänger (become another user)
  - Milliseconds from entry to exit (monitoring)
  - Configurable sleep time (load simulation)
  - Logging/debugging
  - Notify sysadmin of anomalous behavior

PAGS

PAGS At a Glance

- PAGS stands for "Person Attributes Group Service“
- It is a part of the JA-SIG Groups and Permissions subproject (newly available in JA-SIG Subversion)
- PAGS places users into portal groups based on the presence, absence, or value of specified person attributes

PAGS At a Glance (Cont.)

- Use the following configuration files to set up PAGS:
  - properties/PAGSGroupStoreConfig.xml
  - properties/compositeGroupServices.xml
- PAGS groups may have parent groups that originate from other Group Service implementations

Advantages of PAGS

- PAGS groups offer significant advantages in the right circumstances
  - membership does not require management by Portal Administrators
  - status changes are reflected in PAGS groups in real time
  - PAGS groups scale *significantly better* than equivalent database groups

Disadvantages of PAGS

- PAGS groups have a few limitations that make them inappropriate for some circumstances
  - A PAGS group cannot tell you what people belong to it
  - There is no portal UI for managing PAGS groups
  - Must restart uPortal to make a change to PAGS
- This first limitation makes PAGS unsuitable for applications that send emails or post cards, etc.

DLM

DLM At a Glance

- DLM stands for "Distributed Layout Management“
- It was originally developed by Sungard SCT (then Campus Pipeline) for their Luminis product
- Sungard SCT contributed DLM technology to the community, and it's available in uPortal 2.5.x

DLM At a Glance (Cont.)

- Like ALM, DLM has the concept of “Content Fragments”
- Unlike ALM, a fragment is an entire layout (not a tab)
- DLM uses special user accounts called "fragment owners" to define the layouts that make up fragments

Managing DLM Fragments

- Simply log on as the fragment owner to manage the content within a fragment
- The Preferences Channel offers fragment owners the ability to place additional restrictions on movement, modification, or removal by portal users
- Fragment owners and their target audiences may be configured in the properties/dlm.xml file

Running DLM in uPortal 2.5.x

- Despite its popularity, DLM is not the default layout manager implementation in uPortal 2.5.x
- Perform the following steps to convert your portal to use DLM:
  - Adjust 2 properties in properties/portal.properties
    - org.jasig.portal.layout.UserLayoutStoreFactory.implementation
    - org.jasig.portal.layout.UserLayoutManagerFactory.coreImplementation
  - Update the UP_USER_PROFILE table in the PortalDb so that all profiles that used to reference ALM now reference DLM

Popular & Compelling Paradigm

So Why the Fuss?

- The benefits of using these technologies in strategic coordination speak to the core motivations for portal adoption:
  - Personalized, targeted aggregation of content
  - Integration of services and supporting users in their roles

Template Users

- Originally uPortal 2 provided the concept of Template Users to manage content and permissions
- When a user first logged on, all layout content and group assignments would be copied from the template user to the user's new portal account
- Users may be associated with a template user through the uPortalTemplateUserName person attribute
- But the portal has trouble keeping users in sync with their template users

Template Users (Cont.)

- In the case of layouts:
  - If the user gets a different template user, the layout for the new template user *may* be displayed
  - If instead the template user's layout changes, the user's layout *may* reflect the changes
- But neither of these will be possible if the user has made personalized changes to his/her layout

Template Users (Cont.)

- In the case of groups:
  - If the user gets a different template user, the user will be removed from the previous template user's groups and added to those of the new
  - If instead the template user's group assignments change... nothing

Advantages of the New Paradigm

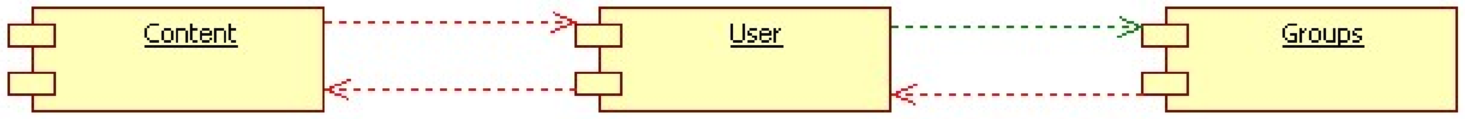
- In the case of layouts:
  - Users will receive new fragments in all expected circumstances
  - Users will see changes to existing fragments in all expected circumstances
- Unlike template users, users can have multiple fragments in their layouts concurrently
- A user who is both 'Student' and 'Staff' can a fragment for each role
- DLM elegantly removes channels that the user is not authorized to view

Advantages of the New Paradigm (Cont.)

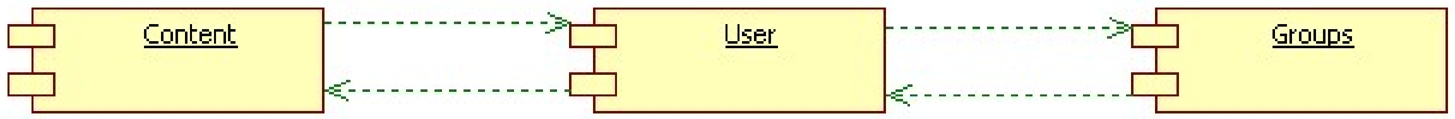
- In the case of groups:
  - If the user's attributes change, his/her membership in PAGS groups will be reflected in real time
  - Likewise if the groups hierarchy changes, the user's position(s) within it will adjust automatically

Graphical Representation

Old Paradigm: Template Users



New Paradigm: PersonDirectory, PAGS, DLM



Advantages of the New Paradigm (Cont.)

- The impact of these differences is significant, and speaks to the portal's "core competency“
  - These features are more robust for targeting content & permissions
  - Using these features requires less hand-on management by portal administrators
  - These features lead to less duplication and risk
  - These features empower portal manages to make more changes, which leads to a more dynamic portal!

Questions?

Andrew Wills

drew@unicon.net

http://objectprincipals.blogspot.com