uPortal Developer's Meeting Minutes

March 11-13, 2002, at University of New Mexico

Monday, March 11, 2002

Attending Members:
Mike Zackrison (Campus Pipeline)
Alex Vigdor (Columbia University)
Steve Barrett (Cornell University)
Mike Oltz (Cornell University)
Michael Ivanov (Instructional Media + Magic)
Justin Tilton (Instructional Media + Magic)
Peter Kharchenko (Interactive Business Solutions)
Ken Weiner (Interactive Business Solutions)
Jim Weaver (Learning Assistant Technologies)
Andrew Draskoy (Memorial University of Newfoundland)
Rachid Drissi (New Mexico State University)
Debra Rundle (Princeton University) [by telephone]
Luis O. Hernandez (Ringling School of Art and Design)
Dave Pickens (Sun Microsystems)
Dave Wallace (University of Delaware)
Mary Berrens (University of New Mexico)
Kelly Cowan (University of New Mexico)
Randy H. Eldredge (University of New Mexico)
Linda Miller (University of New Mexico)
Richard Valdez (University of New Mexico)
Greg Barnes (University of Washington)
Susan Bramhall (Yale University)

I. Release Issues
  A. CVS, Version Numbers
    1. Branches (main, rel-2-0-patches, rel-1-0-patches)
    2. Tags (rel-1-0, rel-1-5, rel-1-6, rel-2-0-alpha, rel-2-0-beta, rel-2-0)
    3. Upcoming tags: rel-2-0-1 and in 6 months rel-2-1
    4. New features and bug fixes should be posted to the "main branch," while bug fixes only should be in the "patches" branch.
    5. Anonymous users allowed to CVS.
  B. Packaging
    1. We currently have the quick-start release (full self-contained version), and "uPortal Only" which is a code snapshot without Tomcat or a database server.
    2. Quickstart version is also convenient for developers to attempt new technologies on a simple installation of uPortal.

3. Need comments on minimum version requirements. Tomcat 4.0.2 has issues related with XML class loading from within the “webapps” folder. The best solution would be to place “jar” files within the “[TOMCAT]/lib/common” directory. It was also mentioned that Tomcat has a configuration file that specifies which classes to avoid loading.

C. Bugzilla

1. Captures bugs and enhancement requests for uPortal.
2. If Bugzilla users receive meaningless reports, an attempt to contact the bug reporter should be made to expand on the comment.

II. Channel CVS Repository (Delaware):

A. Repository

1. Create directory scheme for each institution.
2. Repository will serve public code for CVS channels.
3. Channel applications not integral to the portal should go in the Channel repository. Candidates are (Calendar Channel, UBC Webmail Channel). Webmail Channel needs revision for uPortal 2.0. Note to contact UBC.
4. The uPortal project has been posted in Freshmeat. Username “Jasig”, password “destin”. (http://www.freshmeat.net/projects/uportal) The purpose is to drive more traffic from the outside community to the uPortal project.

B. Issues

1. Need to specify how to check out channels to match the user’s release.
2. May want to parallel the branches from the main CVS to the branches of the Channel repository.
3. CVS for uPortal, Clearinghouse, and CVS for channels will be available at the same time.
4. Projects in the Clearinghouse should be functional. If not, the owner should be contacted. E-mail Dave Wallace dwallace@udel.edu for issues with the Clearinghouse.
5. All dependencies specific to a project should be included with the project.

III. Discussion Lists

A. Needed Features

1. Setup & Administration List.
2. Search feature.
3. Developer list to be made public (read-only). The archives will be available from the JA-SIG website. The traffic in the developer’s list is for main issues related with developing the project. Other discussions should be kept to the general list. If list is to be made public, a moderator will be necessary. Ken Weiner decides who can post to the developer’s list and will accept new members that demonstrate a commitment to contribute.
4. Channels List. Channel Development List deemed not necessary at the moment.
5. Improved FAQ list would be beneficial to the project.

IV. Documentation and the uPortal website

A. Website

1. Current website has proven to be effective in delivering information to both users and developers.

2. The FAQ section could use some more information on implementation and how to use uPortal. The FAQ needs to be expanded and updated for uPortal 2.0.
3. Bill Brooks currently updates the Documentation section of the uPortal site. However, help is needed to continue development. A call is made for someone who can own and organize the content of the section.
4. Pipeline representative stepped in to offer a running effort in Pipeline to document sections of uPortal. He did mention that it was not a solution to our problem, but that it could help.
5. A paid resource is available for the purpose of documentation (technical writer). However, documentation has still not been developed.
6. Jim Hall (Minnesota) may prove to be a resource in developing or organizing documentation. Ken Weiner will contact Jim and discuss.
7. Susan Bramhall will contact Lea Novak (technical writer) to serve as a bridge in developing the content for documentation.
8. Diagrams (database, relational, etc) are not available at the moment. However, presentations are available from previous meetings that could prove helpful in developing diagrams. Susan Bramhall suggested that Lea use the presentations as a good starting point for documentation.

V. Caching

A. What gets cached in uPortal?

1. Channel content should be cached to avoid XML transformations more than once.
2. ICacheable can be implemented by channels to cache content. The class is found in org.jasig.portal.

B. Where is the caching configured?

1. In uPortal properties file there are several properties to turn “on” or “off” caching capabilities for stylesheets, precompiled stylesheets, character caching, size of XSLT cache, and the size of character block cache.
2. Advantage of caching is that page download and processing time on the server side is decreased. However, a disadvantage is that the uPortal may have to be restarted for updated packages to take effect.

C. Issues

1. Caching in uPortal is conglomerated (all caching is done in one blob of memory).
2. A better caching mechanism is needed to maintain separate cache levels for each resource in uPortal.
3. Careful considerations should be given to sensitive information when caching user data.
4. IMultithreadedChannel is available for implementing channels. However, it has not been documented to developers on when or why to use IMultithreadedChannel. It was mentioned that a channel is harder to write using IMultithreadedChannel. GenericXSLTChannel implements but does not extend the BaseMultithreadedChannel.

VI. Properties

A. XML vs. properties formats

1. The general consensus was that the properties format has been rather simple to work with and should not be changed at the time.

B. Consolidation

1. Consolidated much configuration into portal.properties in uPortal v2.0. All properties required to run the system should be placed in portal.properties.
2. Security configuration will be consolidated into portal.properties in the future.
3. Grouping should be done to make the properties file easier to read. For example, the Caching properties in the portal.properties should be grouped so it is easier to find and read.
4. Services.xml file in the properties folder was explained as a file that the portal can use to "kick off" a specific class file at startup. It is used to start off services at start time.
5. Workers.properties file in the properties folder was explained as a file that the portal can use to create "workers" which service the function of responding to streams in the portal. The worker can do anything it wants with the stream. Once a stream is handed to a worker, the worker is responsible for responding to the client. Useful when you want to access a separate servlet but need all the functionality of the portal.

VII. Thread Pool, Viginia Tech Suggestions

A. Why do we need a thread pool?

1. A thread pool will manage threads in the portal, therefore decreasing the cost of starting threads.

B. Issues

1. Thread timeout and synchronization.
    a. Workaround to deprecated "stop" method is to interrupt, and if the thread does not interrupt, then kill it.

C. Implementation Chad La Joie, Virginia Tech

1. Created a thread pool manager that would interrupt the thread if it does not want to give up a lock, however the thread is not killed at any time.
2. Comments are welcome and requested from Sun Microsystems as what the best way would be to synchronize threads and taking care of non-yielding locks. (Sun currently has a document explaining the issue of thread stops at http://java.sun.com/products/jdk/1.2/docs/guide/misc/threadPrimitiveDeprecation.html.)
3. A discussion is to be started on the list as to how best address this issue.

VIII. Administration

A. Channel Administration

1. The need to read Metadata while publishing an RSS channel was discussed. The topic needs to be discussed further in the future when more information is gathered from the users to determine the specific functionality required.
2. A new CEmbededObject channel is desired for media channels.
3. Groups/Permissions slave channel needs to be integrated into Channel Administration.
4. Channel Preview (for both subscribe and publish) is needed in the 2.0 portal version. Channel preview will be integrated in the future. At the moment, a channel can only be rendered once a user subscribes to it. Issues related with channel preview include: user data needs to be used in the channel, channel is outside a layout, etc.

B. User Administration

1. Change password application was discussed but dismissed since uPortal uses different security contexts and authentication methods.
2. Application needed for user maintenance (add/remove, change user attributes such as groups, etc.)

C. Runtime Administration and Statistics

1. Output currently authenticated users.
2. User visits and hits (Channel usage).
3. Memory usage for JVM.

D. Layout Administration

1. Suggest size of columns to users. Maybe enforce 100% addition of all columns.

IX. Usability Issues

A. Internationalization

1. International users are interested in internationalization. Reviewed needs for internationalization on XSLT and determined that it takes a fair number of conditionals and variables that have different languages and a database that has a whole set of different strings for reuse in different languages.
2. Java requires a locale file to be placed for java classes to become internationalized. The process seems straightforward.
3. Internationalization seems straightforward. However, it is a process that will take time, a lot of issues to consider, and will not be an instant implementation. At this point, a retrofit of uPortal would be necessary to implement internationalization.

Respectfully submitted by:

Luis O. Hernández
Ringling School of Art and Design, Webmaster
e-mail: lhernand@ringling.edu

Tuesday, March 12, 2002

Attending Members:
Mike Zackrison (Campus Pipeline)
Dan Ellentuck (Columbia University)
Alex Vigdor (Columbia University)
Steve Barrett (Cornell University)
Mike Oltz (Cornell University)
Peter Kharchenko (Interactive Business Solutions)
Ken Weiner (Interactive Business Solutions)
Jim Farmer (Instructional Media + Magic)
Michael Ivanov (Instructional Media + Magic)
Justin Tilton (Instructional Media + Magic)
Jim Weaver (Learning Assistant Technologies)
Andrew Draskoy (Memorial University of Newfoundland)
Rachid Drissi (New Mexico State University)
Luis O. Hernandez (Ringling School of Art and Design)
Dave Pickens (Sun Microsystems)
Dave Wallace (University of Delaware)
Mary Berrens (University of New Mexico)
Kelly Cowan (University of New Mexico)
Randy H. Eldredge (University of New Mexico)
Richard Valdez (University of New Mexico)
Greg Barnes (University of Washington)
Tom Vreeland (Virtual Education Space)
Susan Bramhall (Yale University)

I. Web Services and uPortal (Jim Weaver, LAT and Ken Weiner, IBS)
- A. Background
  1. "The Possibility Network" from Learning Assistant Technologies, Inc. Software development company specializing in the education space with Java and Internet-enabled technologies.
  2. Three pieces: A corps of learning consultants, store consultants, and learning assistants.
- B. System
  1. Rich client Java Application that updates itself once installed. One software release about every five weeks.
  2. System allows integration with current school information systems (registration, financial aid, and other services.)
  3. Multiple views available for each profile.
- C. Technologies
  1. Java Rich Client offers interactivity, auto-update features.
  2. Enterprise Java Beans implemented on Weblogic Application Server.
- D. Goal with IBS

1. Want to be able to render uPortal Channels directly in PLA system.
2. Joint effort between IBS and LAT.
3. Enable individual portal channels to be accessible via web service.
4. Enable uPortal to be a client to the web service.
5. Web Services interface and channel client will be donated to uPortal project.

E. Architecture
1. uPortal Channels will be interfaced to web services using SOAP.
2. Process of funtionality
    a. Authentication – need more information on SOAP authentication.
    b. Channel instantiation using “fname” and “instance identifier.” The client will be able to make subsequent calls using “fname” and “instance identifier.”
    c. The channel will be rendered with instance identifier, runtime data, and channel markup.
    d. The channel will be destroyed (instance identifier, logout.)

F. Discussion
1. Further discussion will be necessary as we continue with the joint effort to improve on the concept and choose applicable technologies.

G. Already tested
1. Web services communication using Apache Axis.
2. Concept is similar to WSRP (Web Services Remote Portals, proposed standard) from IBM. See http://www-106.ibm.com/developerworks/webservices/library/ws-wsxl
3. Timeline is not set yet before the component is checked in to the CVS. However, it will be about a month or so before the project is mature enough for preview.

II. Distributed Layout Management (Mike Zackrison, Campus Pipeline)

A. What is it?
1. Platform II, Content Management Suite, Integration Suite. uPortal Framework provides the presentation layer to the entire system.
    a. uPortal is integrated into the platform running from iPlanet Web Server for user, group, role, access control, data which is synchronized with the SIS.
    b. Links system to existing applications (webmail, calendar, group and course portals, announcements, and system administration.)
    c. Pipeline has modified uPortal’s default XSLT templates for consistent look and feel and integration with iPlanet and other applications.
2. Distributed Layout Management
    a. DLM provides ways to render a section of a user’s layout into another user’s layout. A user can even allow another user to edit sections of the layout that the other user is allowed to view.

B. How does it work?
1. Administrative look
    a. In lieu of an administrative UI, all DLM definitions and rules are contained in an XML file (dlm.xml in uPortal properties file)
    b. Process
        i. Administrator sets up dlm.xml

ii. Create edit and manage layout fragments and determine required UI elements and restrictions (to administer layout) based on Iperson attributes.
iii. User logs in and sees their composite view, which is comprised of their personal layout fragment, combined with an incorporated layout fragment.

2. System view
    a. Layout is requested.
    b. Personal layout fragment requested from DB.
    c. PLF integrator uses DeleteManager to apply the stored delete directives from PLF to ILF. Uses position manager to process PLF, ILF, and create composite view.

C. Next Steps
1. When code is complete, would like to place it in a globally accessible website. However, if the community sees necessary can provide code earlier.
2. Generalization of code so it works with other styles and profiles.
3. Incorporate model refinements.
4. Create administrative UI channel.
5. Contribute to code base when desired.
6. Request help from uPortal Developers in generalization and model enhancements where possible.

III. Open Source Calendar Server (Greg Barnes, University of Washington)

A. Proposal
1. Not a uPortal-only project.
2. Solve all calendar related problems.
3. Calendar events grouped as follows
    a. Private
    b. Public: official and unofficial
    c. Group: standing and ad hoc

B. Current Problem
1. Calendars are not unified and not easy to find.
2. Packaged calendar solutions are expensive and not customized to specific needs.

C. Proposed system (Scope)
1. Personal calendar
2. Public entry/views
3. Group entry/views
4. uPortal Integration
5. Aggregator
6. Migration/sync (with Palm and other mobile devices)
7. “Add this to my Calendar” and “Subscribe to this” features.
8. Will use existing standards (see III.D.4).

D. Next Steps
1. Create architectural diagram to better explain vision.
2. Will develop and release an Event Store first. IETF standards assume that you have many event stores (iCal, xCal, and vCal are three common standards.)

3. For information on the proposed calendar, go to http://myuwtest.u.washington.edu/uwcal/
4. For information about IETF standards, go to http://www.imc.org/ietf-calendar/index.html
5. Request to download the current version, install, test, and submit bug reports.

E. Current Status

1. A developer's only version is available for download and install (pre-alpha.)
2. The system is database dependent and should work with most standard databases (the development system works with Informix.)

IV. Groups and Permissions (Dan Ellentuck and Alex Vidgor, Columbia University)

A. Overview of Framework

1. Groups and Permissions are actually two separate frameworks, but constantly work with each other.
2. Created out of the need for a framework that was more closely related with a university structure.
3. JNDI is the closest relative to the Columbia Framework (contexts and subcontexts concept.)
4. Perform authorization by attaching permissions to group members. A permission knows about principles, activities, and targets.
5. Allows implementer to plug in security providers.

B. Next Steps

1. Groups do caching currently on a group-by-group basis. We need to extend that into the global cache so groups are retrieved once only.
2. Create a notion of "intersection groups." This would make the search for group membership less resource intensive and would allow for extra features added to the system (ie. If a user is a member of group x and group y, then grant permissions.) The "intersection groups" notion can create confusion with group management.

V. CWebProxy Channel (Andrew Draskoy, MUN)

A. Description

1. Multithreaded channel that allows rendering of third party application output. The third party application thinks it talks directly to the user, but instead talks to uPortal and in turn uPortal renders the output to the user. The third party application trusts that the user has been successfully authenticated in uPortal.
2. Channel was initially a hack but evolved into a full-blown channel service.
3. Need comments on performance testing with stylesheets.

B. Current Status

1. Currently testing caching under default circumstances.
2. Added default parameters into the portal.properties file. (DefaultTimeout, CacheDefaultTimeout.)
3. There are several caching modes: none, all, http:
    a. Several standards are available to implement caching. However, not may parties have implemented the standard.
    b. We don't want the proxy to become a web browser within uPortal.

C. Next Steps

1. IPerson to be implemented so user data is handed over to the third party application.
2. Buttons to be considered in a forthcoming release (pending on uPortal functional button progress) are “Back,” and “Refresh” buttons.
3. For more information, go to http://www.mun.ca/cc/portal/cw/.
4. Further discussion to be started in the developer’s list.

VI. Profiles (Peter Kharchenko, IBS)

A. Description

1. Initially constructed as a basic collection of settings for a user depending on different browsers and user sessions. It enables the user to reach uPortal from different browsers and mobile applications.
2. Contains information about structure, theme transformation, and user layout. The grouping and information is mapped to the user agent.

B. Profile Selection

1. Done using HTTP Headers and Browser Client with hard-coded mappings.
2. The administrator can design a set of regular expressions that match the agent string.

C. Issues

1. There was a general question as to whether LAYOUT_ID in UP_USER_LAYOUT is used by the profiles system. Peter explained that the LAYOUT_ID is not used for “system” users. Ken will make the default value “NULL” since the column is not used at the moment.
2. The User Profile section may overwhelm regular users since it is pretty complex. Some institutions may opt to disable the feature on release.
3. Mike Zackrison (Campus Pipeline): Profiles was turned off in uPortal 2.0 for Pipeline because it was found to solve more of a technical requirement than a user requirement.

VII. Code Maintenance (Peter Kharchenko, IBS)

A. Unit Testing

1. Different kinds of unit testing as defined in http://jakarta.apache.org/cactus/index.html:
    a. Type 1 : code logic unit testing. Probably the best strategy for these tests is to use a Mock Objects type framework.
    b. Type 2 : integration unit testing. Cactus is typically in this category (I'll let you judge if it is the best or not :)). These tests will exercise the interactions with the container.
    c. Type 3 : functional unit testing. These unit tests will let you test the returned values from your server code. This is for example HttpUnit (Note that HttpUnit also performs standard functional testing - as opposed to functional unit testing -, which let you test full use cases - a login use case for example, which is comprised of several requests/responses).
2. A number of different libraries are available to create mock objects. Cactus is a test framework to test server-side Java code (Servlets, EJBs, Tag Libs, Filters, etc.)

B. Refactoring

1. Refactor UserLayout into an abstraction.

2. Cache switching mode.
3. ChannelManager (channel lookup, instantiation).
4. Classes that do too much work should be split so that new features can be added easily and the code is more readable.

Respectfully submitted by:
Luis O. Hernández
Ringling School of Art and Design, Webmaster
e-mail: lhernand@ringling.edu

Wednesday, March 13, 2002

Attending Members:
Mike Zackrison (Campus Pipeline)
Dan Ellentuck (Columbia University)
Alex Vigdor (Columbia University)
Steve Barrett (Cornell University)
Mike Oltz (Cornell University)
Peter Kharchenko (Interactive Business Solutions)
Ken Weiner (Interactive Business Solutions)
Jim Farmer (Instructional Media + Magic)
Michael Ivanov (Instructional Media + Magic)
Justin Tilton (Instructional Media + Magic)
Jim Weaver (Learning Assistant Technologies)
Andrew Draskoy (Memorial University of Newfoundland)
Rachid Drissi (New Mexico State University)
Luis O. Hernandez (Ringling School of Art and Design)
Dave Pickens (Sun Microsystems)
Dave Wallace (University of Delaware)
Mary Berrens (University of New Mexico)
Kelly Cowan (University of New Mexico)
Randy H. Eldredge (University of New Mexico)
Richard Valdez (University of New Mexico)
Greg Barnes (University of Washington)
Susan Bramhall (Yale University)

I. New Features
A. JA-SIG Status (Jim Farmer, IM+M)
1. Building a liaison between dept of Education and Financial Offices.
2. uPortal is beginning to “get a name for itself” and is becoming a respected product.
3. Next steps are much more difficult since it further integrates the portal with educational institutions. This distinguishes the product from commercial implementations. UBC’s mail and Yale’s authentication scheme are great examples of a response to university requests.
4. XML needs to become an integral part of uPortal to become respected as standardized tool.
5. Development is unmanageable since each university tries to implement changes customized to their institution. Local university needs have to be satisfied first since developers are “hired to work for their university first.”
6. The Department of Education has come to the conclusion that authentication should be performed on a university level and not by departments. This imposes an implementation of trust relationship. The Immigration and Naturalization Services will require Digital Certificates for authentication of international students beginning on Jan 1, 2003 (will be phased in over a period of six months,

regulations will come out on the October/November months of 2002). They will also start to offer their date exchanges as SOAP.

7. Three groups working on authentication: libraries, transcript people (California Community College, Texas, Florida), and Federal Financial Aid and Lenders. Jim will recommend to the JA-SIG board that they take on the role of authentication issues. Federal agencies are working in E-Authentication so that each citizen has only one username/password for all Federal programs on the Internet.
8. Expecting other vendors to use uPortal as Pipeline has integrated them into their product.
9. Things that are important to think about while implementing a project:
    a. If you will implement a project and will be standards compliant, let JA-SIG know so that the project is promoted to the XML Group or necessary parties.
    b. An Open Source project will be of interest to vendors. The Open Source project will mandate how a vendor will implement the release, not backwards. There's often a misconception about this, and it defines how the Open Source movement influences the commercial sector.

B. List of Requested New Features

Feature	Description	Dependency	Difficulty	Priority	Volunteer
User layout permission structure	Manager class to mark permissions for nodes	User layout manager		2.1	Michael I.
User layout manager	New class			2.1	Peter K.
Custom channel controls	Establishing mechanism for channel to manage controls that trigger events			Beyond	
Content management investigation	Investigate general interface for plugging in content mgmt sys			2.1	
Content management integration	Implement general interface for plugging in content mgmt sys	Content management investigation		Beyond	
Workflow service investigation	Look into WMFC and others			2.1	
Workflow service	Look into WMFC and others	Workflow service investigation		Beyond	
Aggregated layouts	Administration of layout fragments by different entities	User layout manager	High	2.1	

Channel Archives	CAR files: Packaging strategy and GUIs to work with these files (Java files, images, documentation on how to load database tables, external resources, documentation on how to publish the channel)			2.1	Alex V.
Channel class loaders	Custom class loaders for channels to enforce “sandbox” security			Beyond	
FNAMES	Come up with naming convention and good description on what these are.			2.1	
DbLoader Extensions	Allow command line arguments			2.1	Andrew D.
Groups caching and DB issues	More sophisticated groups caching and db-level locking			2.1	Alex/Dan
Composite Group Service	Composite group services which talks to different group stores, e.g. LDAP			2.1?	Alex/Dan
Pluggable permission policy	Being able to swap out permission logic			2.1	Alex/Dan
Add groups and permissions channels into uPortal				2.1	Alex
Groups/Permissions channel integration	Channel admin channel (channel categories and subscribe permissions)			2.1	Ken/Justin
Idempotent action URLs				2.1	Peter
Frame support	Allows a tab to be a frame, look into use			2.1?	Mark Boyd

	of layers for Netscape 4.7[1]				
Unit Tests	Look at use of Cactus from Jakarta and Mock Maker			2.1	Everyone
Media Manager	Enhance media manager to describe browsers with more detail in stylesheet lists		Low	2.1	Peter
Back button support			High	2.1?	
CError channel	Auto-restart capability, error report capability			2.1	
Integrate user preferences into layout rendering		User layout manager…		2.1	Justin/Ken
Incremental persistence of layout and preferences	Storage procedure is currently done in one complete pass even for small changes			2.1	
Tree-column structure		Integrate user preferences into layout rendering		2.1	Justin
Web services client channel		FNAME stuff		2.1	Ken
Storage managers	Allow channels to get db connections via JNDI			2.1	Susan B.
Investigate JNDI usage	We need to look into using JNDI context that comes with a container			2.1	Ken/Venky
Connection pooling	Add pluggable connection pooling			2.1	
Store implementations	Store implementations should be able to have their own data source managers			2.1	
Channel output filtering	Example: stripping out HTML elements			2.1	

[1] http://developer.netscape.com/docs/examples/dynhtml/scrollable/

	from channel output				
WML implementation				2.1	John Allen Ken W.
Wrappers for other Portal components	Support Jetspeed portlets and others			Beyond	
Registering stylesheets UI				Beyond	
Skin management	Allow skin lists for different themes			2.1	Ken
Tab/column subscription	Ability to subscribe to tabs and columns (layout fragments)	Aggregated layouts		Beyond	
Give channel control over timeouts				2.1	
Internationalization	Make example of portal content in language other than English, users should have ability to customize			2.1	Justin Peter
Convert all properties to XML			Low	Beyond	
Auto-profiles	Research use for centralized information for browser capabilities			Beyond	
GET/POST info	Channels should be able to query ChannelRuntimeData for GET/POST info			2.1	Peter
Runtime param restrictions	Imposes restrictions on ChannelRuntimeData parameters		Low	Beyond	
Multipart form handling	Allow channels to gain access to the Request input stream			2.1	Stephen B.
Outgoing security infrastructure	Obtaining and attaching person info / credentials to the request for an external URL. This info may need to be encrypted.			Meeting	
CWebProxy	Adding caching,			2.1	Andrew D.

	make multithreaded				
HTTP caching implementation	Build utility that channels can use to implement http-level caching			Beyond	Andrew D.
HTTPS Support	Implement JSSE to allow for HTTPS			Beyond	Andrew D.
Docs for "writing a channel"	Differences between implementing IChannel and providing content via existing channel types			2.1	Mike Z. Bill
Channel development kit				Beyond	
Channel preview	Needed during publishing and subscription			2.1	Justin Ken
"Preview and revert" during layout modification	Need to be able to show a user changes and allow him/her to discard them			Beyond	
Enhance CPDs	Need construct that allows adding of arbitrary amount of parameters, will make CApplet and CEmbeddedObject possible			2.1	Ken Justin
User admin	Add/remove users			Beyond	
Runtime admin	Information about memory usage and active sessions	Multi-JVM synchronization		Beyond ?	
Multi-JVM synchronization	Investigate synchronization across multiple JVMz			Beyond?	
Usage statistics	Counting channel renderings, channel accesses, logins (provide entry points)			2.1	Ken
Column widths	Make sure new columns default to 0 width, add message that a total of 100%			2.1	Ken

	is suggested.				
Usability study	Study of layout management, channel admin, etc.			Beyond	Mike Z. Justin T.
Webmail	Add the webmail channel to channel repository, get it to work!!! ☺ Check out JWMA[2]			2.1	Luis H.
Channel Keywords	Associate channel definition with keywords so that the channel can be located by keyword search			Beyond	Debra R.
Simple XML Transformation	Add ability to publish with an XSL URI in addition to an SSL URI			2.1	Ken
RSS Info	Add a standard info screen that gives info on the RSS URL and feed			2.1?	Susan B.
Hierarchical installations	Distributed aggregated layout fragments (DALF)		High	Beyond	
System-wide channels	Allow use of channels that aren T in a user's layout. Right now, a channel like User Preferences has to be in EVERY layout!			2.1	Peter
Thread pool improvements	Look at ways of killing threads and managing pool			2.1	Peter Chad
GenericXSLT improvements	Remembering certain parameters and passing stylesheet a parameter for when channel is root			2.1	Ken
Subscription channel	Subscription channel that can run as servant and also			2.1	Ken Justin

[2] http://jwma.sourceforge.net/

	completes subscribe parameter				

C. Next Steps

1. Ken will post a slightly modified version to the list and request contributions.
2. Next meeting will be held during August on the East coast.
3. A 2.0.1 uPortal version should be released as soon as the major Bugzilla bugs are addressed.
4. A 2.1 release is desired for mid-September or early October.

Respectfully submitted by:
Luis O. Hernández
Ringling School of Art and Design, Webmaster
e-mail: lhernand@ringling.edu