uPortal Developers Meeting Minutes
University of Delaware
February 20-23, 2001

Attendees:

Maria Mosca, Don Fracapane - Columbia University
Dave Steiner, Mark Washington – Rutgers
Shawn Bayern, Susan Bramhall, Howard Gilbert – Yale University
Michael Barton, Debra Rundle – Princeton University
George Lindholm – University of British Columbia
Carl Jacobson, Eric Abbott, John Laker, Li-Huey Lai - University of Delaware
Jim Farmer – University of Delaware
Justin Tilton – instructional media + magic, inc.
Ken Weiner, Peter Kharchenko – IBS
Tom Vreeland - VES Massachusetts

Following on the heels of a positive uPortal meeting at the JA-SIG conference in Destin, Florida, developers met at the University of Delaware to discuss technical issues surrounding uPortal.

The developers met on February 20, which was set aside as a day to cover three main topics. The topics were; uPortal status and update since December 2000, development status update in the areas of security (Yale) and channel keywords (Princeton) and finally uPortal development issues. The remainder of the week was for the developers to spend time working together on uPortal coding.

Jim Farmer, the recently named uPortal project advisor started out the meeting, soliciting introductions from all those in attendance. John, Ken and Peter had assembled an agenda for the meeting, which will be used as the template for these minutes.

Since December

The first item on the agenda was an update of all that has happened with uPortal since the end of 2000.

uPortal training session – Ken Weiner, IBS

> IBS has created a training curriculum for uPortal. Ken and Bernie Durfee held the first training class at CalPoly. The training included discussion on Java development, XSL editing, and channel development.

Website improvements – John Laker, U of Delaware

> John presented a mock-up of the new uPortal website (designed by Maria Mullen) that mimicked the look and feel of the jasig.org website. Michael Barton volunteered to assist with the development of the new website. http://mis105.mis.udel.edu/ja-sig/uportal/

Additional mailing lists – John Laker, U of Delaware

> John mentioned the creation of new mailing lists to complement the existing *jasig-portal* list. *Jasig-portal* would have the role as the general discussion list for asking questions relating to uPortal set-up, configuration and channel development. *Jasig-announce* is a low volume list; this list is for people who are simply interested in uPortal milestone announcements, such as the release of a new version. A third list was also created specifically for active uPortal developers to carry on high volume, technical discussions concerning coding and development issues.

Mellon Foundation grant – Carl Jacobson, U of Delaware

> Although not on the agenda, the group felt it was important to share information regarding the Mellon grant that the University of Delaware received for uPortal development. The grant is for $770,000 over three years with uPortal being self-sufficient by the third year. The goal is to package uPortal as a turnkey product. The grant allows for a part-time manager (Jim Farmer), one or more full-time coders (Ken Weiner, Peter Kharchenko) and a full time writer. uPortal will continue to be open source and open standard. Discussions with Eduprise and others about productizing and service issues have begun, although this doesn't reflect that Eduprise would be used in that capacity. Carl reiterated that Ken Weiner is the technical project lead and Jim Farmer is the project advisor and responsible for higher-level administrative issues.

Since December (cont.)

uPortal release plans – Ken Weiner, IBS

Taking advice from Jim Hall's Open Source presentation at the JA-SIG, Destin conference of "release often", Ken announced the official release of uPortal 1.5. This version consists of bug fixes; implementation of the roles interface which affects, publish, subscribe and render; better stylesheet control, better fonts. Channels developed under 1.5 should port more easily to uPortal 2.0.

uPortal 2.0 is scheduled for release in May 2001, coinciding with the CUMREC conference. Ken defined the 2.0 release as; having all the functionality of uPortal 1.5,but of a fundamental architectural change. uPortal 2 relies on XML and multiple XSL transformations to provide support for multiple clients (browsers, cell phones). Ken also mentioned that uPortal 2.0 would also consist of "anything else we want to add by May", including additional channels (calendar, announcement). Administrative support tools would also be a plus.

Carl commented that uPortal 2.0 would breed serious content developers.
Jim stated that "lots of eyes" will be watching development and will want to know about the architectural advantages of uPortal 2.0.

Ken continued sharing with the group of John's work on the new subscribe, Justin's work on stylesheet development and Peter's work on user preferences and just about every other framework issue.

Ken reported what is missing from uPortal 2.0 development. These items include, finishing user preferences, publish, port of existing security model (*since completed*) and the 1.5 roles interface complete with "hooks into the existing stuff".

Nice to haves for 2.0 would be a simple caching service and administration tools.

Working demo of 2.0 – Ken, John, Peter, Justin

Ken showed the layout management channel, Peter discussed user preferences issues, John showed his subscribe channel and Justin demonstrated the power of XSL for creating sophisticated and versatile new look and feel for uPortal.

Developer Status Update

Security – Shawn Bayern, Yale

Jim opened discussion of security with the potential need for three types of authentication, local authentication, remote authentication (inter-institutional) and aggregation (approximating single sign-on).

Jim requested clarification of Yale's contribution in the area of security for uPortal 2.0. Specifically,

1) Can authentication code which exists in uPortal 1.0 and 1.5 be ported to uPortal 2.0
2) Does this code support aggregation as it stands today
3) If so, what needs to be accomplished in order to move forward with this goal within a July 1, 2001 time frame.

Shawn's response:

1) Authentication code which exists in uPortal 1.0 and 1.5 CAN be ported to uPortal 2.0
2) The existing code does support some types of aggregation but does not include the ability to store credentials permanently for ongoing use. Rather credentials are always acquired at login time and may be cached in memory for the lifetime of the session. Shawn mentioned he is concerned about aggregation because users may "take security for granted" and do not know what they are revealing and to whom.

Yale has developed a new resource dubbed the "Central Authentication Service (CAS)". This tool will be/is being used at their campus to help provide single sign on for web applications. It could also help uPortal provide the same service. Pending Yale's General Counsel approval of a licensing statement, CAS will be contributed to the JA-SIG Clearinghouse (http://www.jasig.org/).

More information regarding the Central Authentication Service can be found at http://www.yale.edu/tp/auth/

Shawn and others propose writing a series of Authentication Adapter classes, stating that users need concrete examples of security implementation. This would assist in the effort of productizing uPortal.

Developer Status Update (cont.)

Channel keywords – Debra Rundle, Princeton

Debra presented the need for the uPortal framework to provide a robust mechanism for content management by the inclusion of keywords.
Statement of the problem:

1) Categorization of channels. As the number of channels at a site increase, the ability to find the channel you want or need decreases.
2) While a portal creates a completely personalized view of web applications and content, the ability for the help desk to manage this flexibility becomes unmanageable. A keyword search would allow help desk personnel to locate channels and give appropriate instructions for use. This search also could assist portal users in searching for and subscribing to appropriate channels.

Keywords related to a channel will be added at channel publish time. Channels may then be searched for either for subscribing to a channel or single channel viewing. At present, a channel must be republished to change keywords.

Development Issues

IChannel API – Peter Kharchenko, IBS

UBC proposes “nailing down” the Channel API to solidify the contract between developers and the framework.

Peter discusses several needs/fixes/updates for the IChannel API. Items include:

Destructors – Clean up. There are at least three areas requiring cleanup, which could be handled via destructors.

- After Render - cleaning cache
- After session - invalidate session, unless we no longer use the session. See Removing session
- After unsubscribe - database cleanup

This led to a discussion of creating “Events” functions. Classifying the events into two categories, Layout_Events and Destruction_Events

Development Issues (cont.)

<u>IChannel API – Peter Kharchenko, IBS (cont.)</u>

Exceptions – Discussion revolves around the need for an ErrorChannel that renders in the place of a channel that is unable to render (ie RSS channel unavailable). Complete with a "try again" option. Peter wanted to define the Channel Exception that would call the ErrorChannel. Exceptions include: Missing Resource (site down); Authorization (no longer permitted to access channel); Rendering problem (SAX exception); Timeout (channel can control it's life). UBC suggests a second use for the ErrorChannel is to support administrative removal of channels. Throw ErrorChannel when user signs into portal, explaining administrative removal of channel. Eliminates the need for utility to parse every users portal_xml and remove dead channel.

Channel properties – What properties should be part of the API and what should be handled by the stylesheet?

- API should handle – isEditable, hasHelp, About (suggested new property)
- Stylesheet handles – isMinimizable, isMaximizable, isDetachable

There were concerns raised regarding the isDetachable property and problems with management of state, and multiple instances of the same channel. Questions were raised whether it should continue to be a supported property. Consensus was that it should be.

Sending files – How does uPortal support the need to send and receive files in a channel? Discussion stemmed from UBC and implementation of attached files in their Email channel. UBC will be investigating the topic to meet their own needs and will present their work in the future.

Removing session – To eliminate the need for session object, what does the API need? Discussion revolved around creation/enhancement of IPerson object to potentially include: request, response, userName, security context, a portal generated PortalSessionID, StyleSheetSet (redesigned for client/browser-type)

Channel registry – Tool(s) for channels to communicate with one another. How do we implement a channel registry? UBC may have some ideas that they can present for further review.

Study other Channel APIs – How do other portal services (Oracle, Jetspeed, Epicentric, Blackboard) implement their channel API? Is there anything uPortal can gain from these implementations? This was left as an open topic for people to research and comment back to the group.

URL semantics – Discussion concerning, user layout root, stylesheet parameters and attributes. Introduce and define a language to discuss the various components

Development Issues (cont.)

Profile – Peter Kharchenko, IBS

Discussion concerning supporting multiple configurations for a uPortal user. The Mime-type describes the markup language to use; from the browser we need the name and version. How do we get the supported features of a client, where does the information come from (cookie, table, detect)? What are the significant "pieces" to define a client and their hierarchy? Mime-type, browser-name, version, platform were suggested as the pieces and their order of significance. Peter expressed concern about how could we maintain a potentially huge list of variations. The consensus of the group was that the current implementation that handles mime-type and browser-name was sufficient for uPortal needs.

Header/Footer

Should the output be after Structure or after Theme transform? Should each item be a channel? Theme transform would handle image processing (graphics) better.

CSS

Do we need a third transformation using CSS stylesheets? This would only be good for "standard browsers".

Justin suggests the use of an XSL:include in the 2nd transformation. He has done this with some of the styles that he has worked with.

I have a comment in my notes; the 2nd transformation has control of CSS params that are passed through for channel development and the classes should be documented so a developer can decide to implement or not. They get a better look and feel if they do implement.

Consensus: "Do it Justin's way".

Publish/Subscribe design

Publish will require a multi-step process. Define the elements/parameters of the channel, define them at publish as to whether they are configurable.
In Layout Manager – add edit channel capability.

Development Issues (cont.)

Security

This topic was deferred until the end of the day to ensure that the rest of the topics were covered. The quick take is that Yale's existing contribution to uPortal 1.0 is fine and will be ported to uPortal 2.0.

The idea of creating a new subscribe-time authentication parameter was offered up to the group. This would be a userid/password store for use by the channel. The data would be stored separately from the portal and encrypted. An example would be the IMAP channel – user stores their id/password at subscribe time for the channel. Channel uses this information after user has authenticated to the portal. Result: a pseudo single sign on. Maintainability and security of data are obvious issues.

There was further discussion of Yale's Central Authentication Service. Comment was made that backend application services have to be modified to be able to get authentication information from CAS.

The question, "How to use chaining?" was asked. An interface for chaining exists in the uPortal 1.0 code and could be ported to 2.0. Statement was made that chaining is useful for other authentication mechanism besides simple password.

SSL

How does uPortal handle the issue of mixed content (secured and non-secured channels)? uPortal faces the same issues as websites that serve up mixed content. Suggestions included:

- Put authentication behind SSL (See Yale's Central Authentication Service examples) Authentication remains secure (encrypted) content remains open.
- SSL-enable Tabs (or folders) A folder that contains all SSL-enabled content. Minimizes mixed content issues.

Consensus: More input from the developer community is needed in this area!

Development Issues (cont.)

Ownership Issues – Ken Weiner, IBS

Several areas in the uPortal development still need work. The team is looking to for volunteers who will lead take ownership of a particular development area and recruit additional team members. These areas include:

- Caching server
- Factories for all of the interfaces (database, person, role, logging)
- Logging (investigate Log4J and other loggers)
- Exception handling (review all code, implementing throws, instead of try/catch as appropriate)
- DB Script maintenance (support more databases, build scripts, review column names, investigate normalization)
- Documentation
- Channel API (channel registry, study other APIs, report findings)
- XSL for RSS 1.0

Note: See **http://mis105.mis.udel.edu/ja-sig/uportal/aor.html** for an evolving list of uPortal development areas.

Plans

The University of Hawaii, CalPoly and The University of British Columbia all offered to host the next uPortal development meeting. The group decided that the next meeting should be at The University of British Columbia.

The meeting is scheduled for the week of March 26.

Comments

There were mixed feelings among the developers as to whether a day was enough time to cover all the information that was covered.
Some suggestions included:

- Extend the discussion to two days
- Have more frequent interactions, perhaps via teleconference, to reduce the amount to discuss
- Keep things the way they are, one day for technical discussion, more time for developers to collaborate
- Separate the meeting into two tracks
    - A technical track geared to solving specific problems
    - A non-technical track to discuss uPortal architectural concepts and other aspects of uPortal.