Developing a Portal Solution Accelerator with Oracle WebCenter Content and WebCenter Portal

ANDY WEAVER
FISHBOWL SOLUTIONS, INC.

Fishbowl Solutions Notice

The information contained in this document represents the current view of Fishbowl Solutions, Inc. on the issues discussed as of the date of publication. Because Fishbowl Solutions must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Fishbowl Solutions, and Fishbowl Solutions cannot guarantee the accuracy of any information presented after the date of publication.

This Whitepaper is for informational purposes only. FISHBOWL SOLUTIONS MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

© 2016 Fishbowl Solutions Corporation. All rights reserved.

Contents

1 Introduction – Executive Overview

After purchasing WebCenter Suite many organizations want a “quick win” project to provide value to their business. One way to do this is through the creation of an employee-facing intranet. Unfortunately many organizations struggle to carry this out because they lack the internal WebCenter knowledge and development experience necessary to make this happen in a quick and cost-effective manner. It is for this reason that Fishbowl Solutions developed a framework allowing organizations to quickly deploy a portal-based and content-driven intranet site without needing any development knowledge of either WebCenter Portal or WebCenter Content. It is this framework that has come to be referred to as Portal Solution Accelerator.

2 Concept & Overview

The concept for developing a Portal Solution Accelerator framework can be broken down into three main parts:

1. Quick startup time (under 30 days to have a fully functioning site)
2. Easy contribution for users
3. User controlled site structure, no IT involvement in the process

Combining WebCenter Portal and WebCenter Content provides the perfect base for creating such a framework. WebCenter Portal, through the use of taskflows and portlets, provides a modular architecture for building out a site. WebCenter Content with its mature Enterprise Content Management capabilities provides an excellent backdrop for creating an easy to use user contribution experience as well as the robust features to support a content driven site structure.

The development process for the framework falls into a three main areas:

4. Site Design
5. Configuring and Customizing WebCenter Content
6. Site development in WebCenter Portal

3 Site Design

The first part of the intranet development cycle that needs to be tackled is the site design. In fact, even during subsequent rollouts of the framework this step still needs to be undertaken in order to make look-and-feel changes that are customer or business-unit specific.

One of the keys to creating a reusable framework was carefully laying out the site structure. It was for this reason that a user–contribution-driven site structure was chosen. This structure consists of a four-level navigation structure presented as a large overlay menu or mega menu.

The items in this menu structure are individual items in WebCenter Content, and their position within the structure is defined by their metadata.

The four-level structure looks as follows:

1. Site Section

    a.) Category

        i.) Sub Category

            (1) Topic

The site section is the top-level navigation element for the framework. The mega menu is presented on the page as a hover-over and shows the children of a particular site section. These children elements make up the next three levels. Categories are simply named containers for Sub Categories and Topics. Sub Categories are just like Categories in that they are container elements only.

The real meat of the site lies in the Topics. Topics are the main content pages within the site and can reside as sub items of both Categories and Sub Categories in the site structure.

With the structure completely defined as content a single display structure can be created and reused between each successive deployment. Since the structure of the site is completely driven by user contribution, a developer only needs to be involved if the display of the menu structure needs to be changed. Setup time for these sites becomes minimal because business users have control over the content and navigation structure.

4 Configuring and Customizing WebCenter Content

Once the site design is finalized the content server needs to be configured to support user contribution to the intranet. This setup consists of three parts: metadata design, content profile creation, and user interface customizations.

4.1 Metadata and Profiles

To support the content-based site structure, a set of content profiles must be created in the content server. The content profiles provide a directed contribution path for the users in order to alleviate as much confusion as possible and make adding and updating content user-friendly.

As part of the framework, profiles were created for each of the main navigational elements (Category, Sub Category, Topic) as well as for content pages and specialized check-ins. A special Article profile is also used to accompany Topics as content pages within the intranet. Articles are just like Topics except that they have no metadata to tie them into the navigational structure. They must be manually linked from somewhere else on the intranet.

4.2 Navigational Fields

The Profiles for elements that show up in the site navigation have a set of metadata fields specifically targeted for this purpose. Those fields are:

- Site Section
- Category
- Sub Category

The Site Section field is an administrator-defined UCM schema view of site sections that will be available on the site. This is set up as a view, as opposed to a standard option list, to facilitate friendly display names instead of backend-limited character names. This also supports future enhancements and attributes that could be associated with a site section.

The Category field is a custom built drop-down list of all the Category content items that are checked into the content server. When a new category is checked in it will automatically appear in this menu as a pick-list option.. Assigning a category to an item means that within the navigational structure of the site the item will show up under that category.

The Sub Category field is implemented in the same way as the Category field but has the extra distinction of being a dynamic choice list based on the Category that is selected. Selecting a Sub Category is optional if the item does not need an extra level of menu definition.

4.3 Personalization Fields

One of the most interesting requirements for a corporate intranet is the ability to deliver personalized content to users based on their attributes in LDAP. To facilitate this, four metadata fields are presented on each profile that match up with four custom attributes in the company LDAP.

Examples of fields that can be added to personalize on:

- Employee Location
- Employee Role
- Employee Type
- Employee Department

When the content is queried from the portal side, the four user attributes are passed in as part of the query so that the relevant content is returned to the user.

4.4 Related Content

Another requirement that must be fulfilled on the content side is giving contributors the ability to specify items that are related to a particular topic or article page within the site. These related items will then show up to the left and right of the main content item when displayed to the user.

On the content profile for the topic and article pages two fields were added that store a list of content ids of items that are related to that item. A user interface customization (detailed below) was created to provide a better user experience for finding the related content.

4.5 User Interface Customizations

One of the driving concepts behind developing a Portal Solution Accelerator framework was that the contribution experience for the user should be as streamlined as possible. To enable a better contribution experience for the user a set of user interface customizations are included in a component to as part of the framework. The two main customizations that were made are:

- WYSIWYG editor integrated into the check-in page
- Popup dialogs to add values to metadata fields

Integrating a WYSIWYG editor for content contribution directly into the WebCenter Content check-in page removes any confusion for the user as to what content is being contributed to the site. It also eliminates the need to have a file separately stored on the contributor's file system that would be edited independently. The WYSIWIG editor chosen for this integration was the latest version of the ckEditor.

As mentioned above there are several custom metadata fields that drive the display of data on the site. The fields that store which items are related to a specific piece of content can be particularly cumbersome to manage. Because of this challenge, custom popup dialogs were added to the check-in screen for these fields in order to provide users with a way to search for and then select which content items they want to relate to the current item. These selections are then stored in the associated metadata field and evaluated when displaying the page on the portal.

5 Site Development in WebCenter Portal: Spaces

With the site design and structure laid out and the content server configured the next step is to develop the frontend framework for how content will be consumed in the portal environment. The first question that must be answered is whether to develop the site as a custom portal application or build it on top of WebCenter Spaces?

For the Portal Solution Accelerator framework WebCenter Spaces was chosen as the base for the site. This brought with it a slightly higher level of complexity but also brought the following advantages:

- Collaboration features present and easy to immediately integrate
- Portal site framework already in place, full administration
- Support for multiple sites by using group spaces

Under this framework, the intranet resides as a single space within the Spaces application. Since the navigation for the intranet is driven by user content contribution a custom navigation element needed to be developed. This element is included as part of page template in the portal. The page template is then assigned to that space allowing the custom navigation to be rendered for the site.

The last design consideration for developing within the portal was determining what the content pages would look like and how to aggregate all the information. The intranet framework uses a dynamic page rendering approach under which the content is displayed based on the content ID that is passed into it. This keeps the number of pages that need to be created in the portal to a minimum since all of the content on the page is queried for dynamically.

6 Developing the Funcionality

With the design decisions compete and base framework chosen development can begin. The framework code can be broken down into the following main pieces:

- Managed beans
- Taskflows
- Content consumption portlets

Using a combination of these three implementation methods the framework starts to take shape.

6.1 Managed Beans

The managed beans are the backbone of the implementation. They contain all the logic for retrieving content from the content server and assembling it into data structures that can be displayed on the portal page. The framework contains several managed beans to provide the backend for the functionality of the site. The main ones are:

- ContentManagerBean
- SiteSearchBean
- UserProfileBean
- PersonalizationBean

The ContentManagerBean is the main access point for all content related calls for the intranet site. It provides methods for retrieving the menu navigation structure, retrieving content information for a content page, and retrieving the list of related content for a page. Any custom taskflow that is developed as part of the framework makes its content calls through this bean.

The SiteSearchBean handles searching for intranet content. It constructs the query, executes the search using the UCMBridge class, and builds out the data structure to be returned for display.

The UserProfileBean provides access to any custom LDAP object for the logged in user that is not part of the standard WebCenter user profile. It also provides methods for looking up custom attributes for an arbitrary username.

The PersonalizationBean allows for an administrative user to “impersonate” the custom LDAP attributes of a user so that the contributor can test new content as it is being entered. This

alleviates the need to have test users created in the LDAP to match the multitude of possibilities for the various personalization attributes.

6.2 Accessing the Content Repository Connection

In order to easily access the content server the managed beans need to get a hook into the already configured content repository connection. This can be accomplished by leveraging the UCMBridge class. This is the helper class that the content repository calls in the Spaces application and end up getting routed through.

To grab an instance of this class do the following:

- Grab the connection name using the DocLibADFConfigUtils class
- Grab the singleton instance of the UCMBridge class

Example:

```
import oracle.webcenter.content.integration.spi.ucm.UCMBridge;
import oracle.webcenter.doclib.internal.config.DocLibADFConfigUtils;
.
.
.
String                          connectionName                          =
DocLibADFConfigUtils.getPrimaryConnectionName();
UCMBridge ucmBridge = UCMBridge.getBridge(connectionName);
```

Once an instance of the class is obtained content server services can be directly executed using the executeRequest() method.

Example of executing the standard search service to return category data for the menu structure:

```
DataBinder                    categoryQueryBinder                    =
ucmBridge.createDataBinder("GET_SEARCH_RESULTS");
String categoryQueryText = "(xIdcProfile <matches> `intranetCategory`
<and> xSiteSection <matches> `"+sectionName+"`)";
categoryQueryText += generatePznQuery();
categoryQueryBinder.putLocal("QueryText", categoryQueryText);
categoryQueryBinder.putLocal("SortField", "xSortOrder");
categoryQueryBinder.putLocal("SortOrder", "ASC");
categoryQueryBinder.putLocal("ResultCount", "100");
IdcContext               idcContext               =               new
IdcContext(FacesContext.getCurrentInstance().getExternalContext().getR
emoteUser());
DataBinder   responseBinder   =   ucmBridge.executeRequest(idcContext,
categoryQueryBinder);
```

Using the UCMBridge class provides several benefits. The main benefit is that it is the class that Oracle uses on the backend to implement the document library features in the WebCenter Spaces application. It allows the framework to be redeployed across any number of servers or

environments without the need for extra configuration to connect to the WebCenter Content repository, which leads to easier deployment.

6.3 Taskflows

The framework contains a set of ADF taskflows that provide the display for functionality defined in the managed beans. These taskflows display the dynamic information pulled in on a content page based on the content ID. Examples of the taskflows include:

- Local Level Navigation
- Related Content
- Breadcrumb

The Local Level Navigation taskflow queries for content items that are at the same Topic level in the menu structure as the item that is currently being displayed. This provides a simple and intuitive navigation structure on a per–content-page basis that is automatically generated.

The Related Content taskflow displays a list of any related items that are assigned to the content being displayed. This concept can be extended to display something like contact information for information on the page. An example of this would be displaying the information to contact HR on the benefits page.

The Breadcrumb taskflow does exactly what it says—generates a breadcrumb for the current item based on where it is in the menu structure.

Utilizing ADF taskflows for this functionality makes it very easy to reuse them across multiple deployments of the framework. The taskflows can be placed on the content page in a way that can fit almost any page design.

6.4 Content consumption portlets

The final piece of the puzzle is consuming the actual content of the pages in the portal. The UCMBridge class above only provides methods to execute WebCenter Content DataBinder based requests. This means that the GET_FILE service cannot be executed with the bridge class since the content will not be streamed back for display.

To solve this problem prebuilt and reusable portlets were developed to directly stream the content from the content server and display it within the page. These portlets are also leveraged to display complex lists of content on the intranet's home page such asa rotating banner of content items featuring the latest news stories.

Using portlets for this functionality gives the most flexibility in how the content is retrieved and displayed back to the user. It also enables the most configuration options for an admin or site designer to modify a page in the future.

7 Packing it up for Reuse

What would a Portal Solution Accelerator framework be without defining what is actually in the proverbial box? In this case it is a fairly simple set of items that make up deployment of the framework. The deployment consists of the following artifacts:

- WebCenter Spaces shared library extension
- WebCenter Content CMU (Config Migration Utility) bundle
- Portlets EAR file
- Exported sample space and assets

By using the above artifacts a development instance can be set up and ready for contribution within 1-2 days of “opening the box”. This provides the quick start that a lot of companies are looking for with their WebCenter investment.

8 Conclusion

For a lot of companies that have invested in the WebCenter Suite of products getting an application up and running in a short period of time can be a daunting task. Although providing an internal portal or intranet for employees is a common reason for purchasing WebCenter, getting everything setup and developed can take months of work. Utilizing a pre-built framework to kick start a portal-based intranet can help reduce that startup time significantly. Developing such a framework is an interesting and challenging task. Sticking to core concepts like user-driven site structure, reusability, and ease of contribution allow it to be redeployed across many environments. Leveraging the strengths of the WebCenter products line, especially WebCenter Content and Spaces, provides the perfect base for such a framework to be created and reused many times over.