The International Journal Of Engineering And Science (IJES)
|| Volume || 5 || Issue || 2 || Pages || PP -28-40|| 2016 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

A content-Incentive-Usability Framework for Corporate Portal Design from a Knowledge Management Perspective

Hillary Kipkorir Langat, Dr. Cheruiyot W.K, PhD. And Sylvester Kiptoo

Jomo Kenyatta University of Agriculture and Technology, Nairobi CBD Campus , Institute of Computer Science and Technology, P.O Box 62000,Nairobi Kenya

Jomo Kenyatta University of Agriculture and Technology, Main Campus , Institute of Computer Science and Technology, P.O Box 62000,Nairobi Kenya

ABSTRACT

The role of corporate portal on consolidating an organization's internal and external information and knowledge repositories has been recognized by many organizations worldwide including manufacturing firms. The purpose of the study was therefore to create a Content-Incentive-Usability Framework for Corporate Portal Design from a Knowledge Management Perspective looking at Unga Group Limited. It specifically investigated the creation and integration of content-Incentive-Usability Framework for Corporate Portal Design; Experimental research design was used as the research design. The architecture described here realized data integration using EAI systems. These offerred functions for the mapping and transportation of heterogeneous data from the different applications involved. The web service standards SOAP, WSDL and UDDI provided the framework for inter-organizational and cross-application system-to-system communication through remote function calls. While portlets only allow reading access to the data of different heterogeneous applications, EAI systems such as e.g. the IBM Websphere Business Integration Suite or the mySAP Exchange Infrastructure provided coordinated writing access to different systems and their data. The study therefore concluded that there was a created, integrated, developed and Validated Content-Incentive-Usability Framework for Corporate Portal Design for Unga Group Limited and recommends that Unga Group Limited should invest more in content management tools that offer abilities to integrate, classify, and codify knowledge from various sources.

[1] *Masters Student in Computer Systems from Jomo Kenyatta University of Agriculture and Technology*

Date of Submission: 04 February 2016 Date of Accepted: 20 February 2016

I. Introduction of the Background Information

The knowledge based theory of the firm argues that firms obtain competitive advantage by creating, storing and applying knowledge (Javatilaka, Schwarz & Hirrschheim, 2009). According to Grant & Baden-Fuller (2010), a firm’s ability to leverage knowledge held by members in the organization is dependent on first, the ability of the firm to create an infrastructure to access this knowledge, transfer it and make it available to others. A second determinant is the extent to which the knowledge that is captured matches with the product domain of the firm.

Enterprise Information Portals have emerged as gateways to streamline information access in firms (Kim, Chaudhury, & Rao, 2010). The first service they provide is access to transactions with the various information sources scattered across the enterprise, such as structured databases, email servers and document repositories. A second service is access to data and knowledge from both internal and external information sources, such as the world wide web (WWW). Finally, these portals allow users to interact with other users to perform activities that require team collaborations.

Corporate portals provide several benefits for corporations, and their employees, partners and customers. There are several vendors of corporate portals reported by (Raol et al., 2009; Rose, 2009). In the academic world, the majority of academic institutions are using Campus Pipeline; other software that are highly used are Epicentric, Oracle, and Home-grown software according to a recent research on the use of corporate portal in the academic world (Li & Wood, 2011).

A large body of literature exists on evaluating and enhancing the usability of computer systems in general (Nielsen, 2003; Shneiderman, 2008). Typical constructs include the learnability of the system (how long does it take to reach a steady state of proficiency?), the efficacy (error rates made by users when performing benchmark tasks), the efficiency (how quickly can users perform benchmark tasks) and the subjective satisfaction of the user.

This study having considered the strengths and importance of corporate portals for knowledge management in organizations endeavored to create a Content-Incentive-Usability Framework for Corporate Portal Design from a Knowledge Management Perspective looking at Unga Group Limited.

1.2 Statement of the Problem

As earlier noted, to maintain competitive advantage and settle on a workable strategy, organizations are considering corporate portals. The role of corporate portal on consolidating an organization's internal and external information and knowledge repositories has been recognized by many organizations worldwide including manufacturing firms. As a result,there has been much interest in the Enterprise Information Portals which have emerged as gateways to streamline information access in firms. While many frameworks have been considered to help, none has clearly considered a framework that specifically looks into the corporate portal on a knowledge management perspective and considered its content, incenstive and usability. This study therefore hope to fill the gap by creating a Content-Incentive-Usability Framework for Corporate Portal Design from a Knowledge Management Perspective looking at Unga Group Limited.

1.3 Research Objectives

1.3.1 Main Objective

To create a Content-Incentive-Usability Framework for Corporate Portal Design from a Knowledge Management Perspective looking at Unga Group Limited.

1.3.2 Specific Objective

1. To investigate the creation of content-Incentive-Usability Framework for Corporate Portal Design.
2. To investigate the integration of content-Incentive-Usability Framework for Corporate Portal Design.

1.4 Research Questions

1. How will the Content-Incentive-Usability Framework be created in Corporate Portal Design?
2. How will the Content-Incentive-Usability Framework be integrated in Corporate Portal Design?

II. LITERATURE REVIEW

2.1 The CIU Framework

2.1.1 The Content Dimension for KPs

The content dimension deals with the determination of the content that should be presented on the KP (what should be presented) and the process of creation of the content (what are the challenges facing this content creation?). The study subdivides this dimension into the following sub-dimensions: *elicitation and translation of tacit and explicit knowledge, the integration of structured and unstructured data* and *the creation of a knowledge ontology to enhance availability*.

2.1.1.1 Elicitation and translation of tacit and explicit knowledge

According to Nonaka & Takeuchi (2011), tacit knowledge embodies beliefs and values, and is actionable. In contrast, explicit knowledge is codifiable into artifacts such as documents, or multimedia formats. Both are essential for organizational effectiveness.

The transmission of knowledge from one individual to another can take the forms shown in table 1.

Conversion	**Process**	**Facilitating Technologies**
Tacit to Tacit	Socialization	E-meetings, Chat
Tacit to Explicit	Externalization	Chat
Explicit to Tacit	Internalization	Visualization of data
Explicit to Explicit	Combination	Text search, document categorization

Table 1. Conversion of knowledge (Nonaka and Takeuchi, 2011)

Of the possibilities shown in table 1, the elicitation of tacit knowledge from experts, and the codification into explicit knowledge represents an important task in the creation of a KP. Eraut (2012) found that elicitation task was easier if:

- there was a mediating object that experts were used to, such as a drawing, a picture or a graph,
- a precedent of regular mutual consultation existed between novices and experts,
- a training or mentoring relationship was part of the cultural and behavioral expectations in the organization,
- informal meetings were held, where ‘riskier’ comments could be made, and
- there was a perceived potential crisis or change

The degree to which a KP allows the translation of knowledge will influence the final quality of content. Table 1 lists some example technologies that can be used to facilitate the conversions. For example, if we need to capture the tacit knowledge of an expert into a KP we need to make this tacit knowledge explicit, which can be facilitated by conversations with the expert. The explicit knowledge may then need to become tacit within other users in order to transfer the expertise, and this process can be enhanced if the explicit knowledge is presented on the KP in a form that is easy to visualize.

2.1.1.2 Integration of structured and unstructured data

Every organization has a large amount of data scattered in sources such as structured databases, e-mail, documents, blogs and newsgroups set up for specific user groups. A major challenge in constructing a KP is the integration of this information. The use of semi-structured data to integrate heterogeneous data sources has been shown in several works such as (Fernandez, Florescu, Levy, & Suciu, 2012; Garcia-Molina et al., 2011). We characterize the issues that need to be addressed in this integration at different layers: *the physical layer*, *the syntax layer* and *the semantic layer*. This is similar to the approach used in (Jin, Decker, & Wiederhold, 2011) which uses integration, semantic, composition and generational layers.

The physical layer involves the composition of the files that store this data. These files include relational database management system (DBMS) files, word processed documents in various formats and text based or hypertext markup language (HTML) files for email, blogs and newsgroups. Part of the challenge is that in most cases, these “islands of information” are not touched, and an automated integration mechanism needs to be created for real-time updating of the KP from these multiple feeds.

The syntax layer deals with the representation of the same information in different formats. For example, information on the same customer may be scattered and/or duplicated across multiple relational DBMSs, documents, blogs newsgroups and emails. Duplicated information may have different labels, so that one system may use the *customer_id* as the unique identifier, while another may use the *customer_account_number* for the same purpose. The usage of eXtensible Markup Language (XML) (Glavinic, 2010) has greatly simplified the mechanism of automation. However, firms still face the organizational challenge of creating a common XML schema that can be fed from these multiple streams. Examples of existing XML schemas that may be used include the TSIMMIS approach in (Garcia-Molina et al., 1995) for structured data and the resource description framework (RDF) (Jin, Decker, & Wiederhold, 2011) for semi-structured information.

The semantic layer deals with the inference of meaning from the data. We propose that one way to accomplish this is to link the data to processes performed by the end-user of the KP. A second method to accomplish this is to create meta-categories of the data that map to a knowledge ontology. For example, information on customers, purchases, products and promotions may be combined into a “selling assistant” screen that can be part of the KP. In order to create meta-categories, the meaning of the data needs to be understood. The semantic layer feeds into the creation of a knowledge ontology, which is described next.

2.1.1.3 The knowledge ontology in a KP

The question of what defines knowledge needs to be answered if knowledge is to be codified and made available. Examples of knowledge include reports and charts from structured data, summary statistics on unstructured data (such as the number of emails sent to a customer), and data mining into templates (which are part of the ontology) from blogs, newsgroups and documents. The aim here is to match the knowledge ontology to the product domain and the organizational structure of the firm, to increase efficacy of the KP (Marwick, 2011). For example, in a process driven organization, the knowledge ontology may stem from process descriptions that are already developed. In a functional organization, in contrast, the knowledge ontology would be better off incorporating the functional areas such as sales, marketing, accounting, and operations.

Many ways to develop ontologies have been suggested. Some suggestions include using text classifiers (Woods, Poteet, Kao, & Quach, 2006), allowing individual employees to add to an existing list of terms (Amidon & Macnamara, 2009), and forming expert sub-groups of employees to develop key words to be incorporated into the ontology (Markus, 2011). However, using these methods individually to develop ontologies can create problems. In the case of text classifiers, this method only allows for ontologies that use existing documents. It is important to share other forms of knowledge such as lessons learned (Gill, 2011; Holsapple & Jones, 2012). This type of knowledge may not be represented in a documented format at the time the ontology is created and key terms may be missed.

A potential problem of allowing individual employees to simply add to an existing list is the organization may end up with so many “key” terms that nothing can be grouped. For example, if one employee uses the term “business reengineering” and another employee uses the term “organizational redesign” and each added their own term to the list of organizational terms, then the knowledge categorized as “business reengineering” and the knowledge categorized as “organizational redesign” may not be grouped together.

Forming expert sub-groups to develop an ontology may solve the above problem. However, now there is the problem of novices not knowing enough to search for the correct key word (Markus, 2011). If the employees are unable to utilize the system designed to do this, then only those who already possessed the knowledge would use the system.

2.1.2 The Incentive dimension for KPs

Historically, companies have driven their employees to excel through competition (Van Alstyne, 2011). This practice has resulted in employees hoarding their knowledge in order to keep a competitive edge over their co-workers. In this new era of knowledge management (KM), there has been an organizational shift to knowledge sharing. In order for organizations to fully utilize and benefit from the knowledge within the organization, they must find ways in which to encourage employees to share their knowledge (King, 2012). In addition, organizations need to provide means for which the employees can easily participate in knowledge sharing. These activities of securing knowledge sharing efforts and structuring knowledge sharing efforts encompass the knowledge coordination class of activities (for further information on this KM class of activities see Holsapple & Jones, 2011).

Social exchange theory indicates that individuals will only contribute when there is an expectation of some future benefit. According to this theory, organizations will need to find ways to illustrate to employees the potential returns of sharing their knowledge (Markus, 2011). Therefore, practices such as rewarding employees for sharing their knowledge with others (Bose, 2010; Liebowitz & Chen, 2009), describing just how that knowledge sharing can be of benefit at both the individual and the organizational level (Delio, 2008; Department.of.Navy, 2011), publicly recognizing “team players” (Delio, 2008), and rewarding employees for participating in a knowledge community (Smith & McKeen, 2009) are ways which companies may motivate employees to participate.

2.1.3 The Usability Dimension for KPs

A large body of literature exists on evaluating and enhancing the usability of computer systems in general (Nielsen, 1993; Shneiderman, 2008). Typical constructs include the **learnability** of the system (how long does it take to reach a steady state of proficiency?), the **efficacy** (error rates made by users when performing benchmark tasks), the **efficiency** (how quickly can users perform benchmark tasks) and the **subjective satisfaction** of the user. While the first four are clearly measurable, subjective satisfaction can be measured in several different ways. It has been investigated in terms of attitude towards use in many studies (Chou, Hsu, Yeh, & Ho, 2011; Heijden, 2009). Usefulness and ease of use are deeply rooted in attitude towards use. Perceived usefulness is the degree to which users believe that a Web portal will enhance their performance, and perceived ease of use is the degree to which users believe a Web portal will be free of effort (Chou, Hsu, Yeh, & Ho, 2011). Usefulness and ease of use have been found to have a significant impact on a users’ intention to use a Web site (Heijden, 2009; Lin, Wu, & Tsai, 2011).

User acceptance has also been investigated in terms of data quality and knowledge distribution (Chou, Hsu, Yeh, & Ho, 2011). Data quality means the information provided by the Web portal must fit the use of the consumers and generate useful information for the users’ decision-making. Knowledge distribution deals with the need for users’ to use industry Web portals to facilitate employees’ growth and cross-department knowledge sharing. Heijden (2009) also found perceived enjoyment to influence user acceptance of Web portals. Enjoyment is the extent to which using a Web portal is perceived to be enjoyable on its own.

As an example of usability evaluation in the area of web portals, Yang, Cai, Zhou, & Zhou (2011) developed and validated an instrument to measure perceived subjective service quality of Web portals. The instrument focused on five key dimensions of service quality: (1) usability, (2) usefulness of content, (3) adequacy of information, (4) accessibility, and (5) interaction. Service quality can be seen as a dimension of user acceptance. The five measures of service quality can therefore have an impact on acceptance of Web portals. *Usability* is related to user friendliness, and it is primarily identified in terms of layout, Web site structure, user interface, appearance and visual design, clarity, and ease of navigation. *Usefulness of content* is the value, reliability, accuracy, and currency of the information provided by the Web portal, where as adequacy of information is completeness of the information provided by the Web portal. *Accessibility* of the Web portal involves availability and responsiveness of the Web site. Finally, *interaction* exists between the users and service providers’ employees, and users and the Web site, and among peer users of similar products.

III. METHODOLOGY

3.1 Research Design

Experimental research design was used as the research design. This design was necessary to get the CIU framework that was then measured based on its performance for corporate portals for knowledge management.

According to Seltman (2014) this design is relevant as it allows for both graphical and non-graphical exploratory data analysis (EDA). This study looked at the architecture, design, development and validation of the CIU corporate portal.

3.2 CIU Architecture

With regard to client/server architecture, the existing models in the relevant literature and in practice frequently only describe the business management functionality of applications. Portal architectures, however, call for an integrative analysis of the levels presentation, application functionality and data which extends beyond the limits of systems and organizations. The added value of portals, after all, is derived from the backend integration of the application functionality and data offered.

The modules of the CIU architecture will offer the following functions: if a user accesses a portal, it checks the user's authenticity, authorizes access to the integrated applications, e.g. through single sign-on, and regulates communication between client and portal (security).

3.2.1 Integration

Portals integrate the functions of different, mostly heterogeneous applications and place these in the context of specific business processes. Unlike conventional architectures, however, portal architectures integrate applications at the level of the user interface as well as at the level of functionality and data. An interface which is homogeneous but integrated across various applications replaces the specific front-ends of these applications.

The portal integrates the knowledge process-related functionality of knowledge management systems by means of portlets.

3.2.2 Presentation Integration: Portlets

From the point of view of the web browser, portlets are web applications, which are developed using any programming language and exchange data on the basis of standards. Examples range for simple functions such as collecting and sending e-mails to more complex functions such as displaying a parts list or determining all the transactions and sales of a specific customer in a given year.

Several portlets can be positioned per portal page. There are basically two possibilities for integrating portlets into a portal page:

- Browser-side Integration. This type of integration uses iFrames.
- Server-side Integration. Server-side integration is based on Server Side Include technology (SSI). Here, an HTML page is compiled by the web server from a number of individual areas (rendering) and then sent to the browser complete.
- Parsed HTML/XML. Since HTML pages, unlike XML documents, are unstructured, they are only of limited suitability for structured data processing, as to be found, for example, in databases.

3.2.3 Function Integration: Web Services

A major design element of integration architecture is web services. From a technical point of view, web services are aimed at communication between loosely linked software components. The web service standards SOAP, WSDL and UDDI will provide the framework for inter-organizational and cross-application system-to-system communication through remote function calls.

- Data-oriented web services provide XML documents as output.
- Presentation-oriented web services incorporate presentation and interaction logic.

3.2.4 Data Integration: Enterprise Application

While portlets only allow reading access to the data of different heterogeneous applications, EAI systems such as e.g. the IBM Websphere Business Integration Suite or the mySAP Exchange Infrastructure will provide coordinated writing access to different systems and their data.

Web services and EAI are mutually complementary rather than competing approaches. Whereas EAI systems will be used primarily in the intra-organizational field, the high level of standardization in the case of web services will make them suitable for inter-organizational applications. Here, the use of standards which are accepted at an inter-organizational level can obviate the need for proprietary EAI adapters.

IV. RESULTS

4.1 creation and integration of content-Incentive-Usability Framework for Corporate Portal Design

The main dimensions and sub-dimensions of the CIU framework are summarized in figure 1

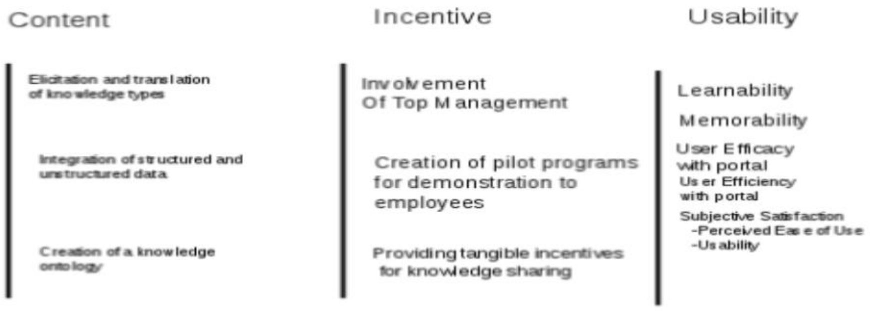

Content
Elicitation and translation of knowledge types
Integration of structured and unstructured data
Creation of a knowledge ontology
Incentive
Involvement Of Top Management
Creation of pilot programs for demonstration to employees
Providing tangible incentives for knowledge sharing
Usability
Learnability
Memorability
User Efficacy with portal
User Efficiency with portal
Subjective Satisfaction
-Perceived Ease of Use
-Usability

Figure 1. Dimensions of the Content – Incentive - Usability Framework

The CIU framework can be utilized in several ways. From a practical perspective, it serves as a checklist for organizations who are exploring the implementation of a KP. The discussion of each of the sub-dimensions in this work should provide prescriptive guidance on increasing the impact of the KP on the performance of the firm. Thus, focusing only on the content without providing incentive or making the portal usable may reduce the chances of success. A 3-pronged approach that addresses all three dimensions will increase the potential impact of the portal.

From a theoretical standpoint, the CIU framework serves to provide perspective in the different areas of research related to KPs. Thus, future research projects can be more easily put into perspective with other work, by utilizing this framework to align the project with a particular dimension and sub-dimension.

4.1.1 The general Process of the creation of Content-Incentive-Usability Framework for Corporate Portal Design

1. **Relevant Repository Framework APIs and documentation**

A prerequisite for developers who are enhancing a Repository Manager implementation is an understanding of the concepts described in Repository Framework Concepts (RCO) and an understanding of Implementing a Repository Manager (read-only) (REP). It is also recommended that you have a basic understanding of Repository Manager implementations. This section describes the usage of the following repository framework APIs that was relevant for implementing the Content-Incentive-Usability Framework for Corporate Portal Design

- com.sap.netweaver.bc.rf.common.*: Contains repository framework common interfaces
- com.sap.netweaver.bc.rf.mi.*: Contains repository framework manager interfaces that are necessary for the implementation classes

Considerations for Repository Manager Enhancements

The enhancement described in this section is based on a mutable Repository Manager implementation which uses a ZIP archive as a “backend” data source. The ZIP archive has to be stored on the portal server on which the RM is deployed.

Read/Write mode

For introducing the read/write mode, the mutable interfaces for the mandatory aspects of namespace, content and property was taken as an example and are shown in the Coding in Detail section later on. The coding example also contain mutable interface implementation for all other aspects mentioned in this section.

Security aspect

The security aspect was handled from a Security Checker point of view. The standard KM AclSecurityManager was used as an example configuration for Security Managers. There was no explicit custom based Security Manager in this part.

Coding in Detail

Implementing a repository manager meant implementing certain unified aspects. The minimum requirement was that ones implement the namespace, content, and property aspects. This example dealt with the mandatory aspects and enhances them with advanced aspects like security checks and locking.

Read/Write mode

To enable the read/write mode on a repository, the corresponding repository manager implementation had to be extended by its supported option set.

For the namespace aspect the following options have to be added to the set:

```
supportedOptions.add(com.sap.netweaver.bc.rf.common.namespace.SupportedOption.CREATE_COLLECTIO
N);
supportedOptions.add(com.sap.netweaver.bc.rf.common.namespace.SupportedOption.CREATE_RESOURCE);
supportedOptions.add(com.sap.netweaver.bc.rf.common.namespace.SupportedOption.DELETE);
```

This indicates that this implementation supported the read/write mode for creating collection, creating resources and deleting resources.

In parallel to the content and property aspects, the following options had to be added:

```
supportedOptions.add(com.sap.netweaver.bc.rf.common.property.SupportedOption.SET_PROPERTIES);
supportedOptions.add(com.sap.netweaver.bc.rf.common.content.SupportedOption.UPDATE_CONTENT);
```

This indicates that the implementation supported the read/write mode for setting/updating properties and create/update content.

Read/write mode for other aspects, e.g. locking, had to be added according to the scheme above.

Namespace Manager Aspect

To enable the Namespace aspect with the read/write mode the developer had to implement the IMutableNamespaceManager interface. This interface extends the immuntable (read-only) version of the INamespaceManager interface. It was still necessary to extend the AbstractSubManager class for using the RF object lifecycle.

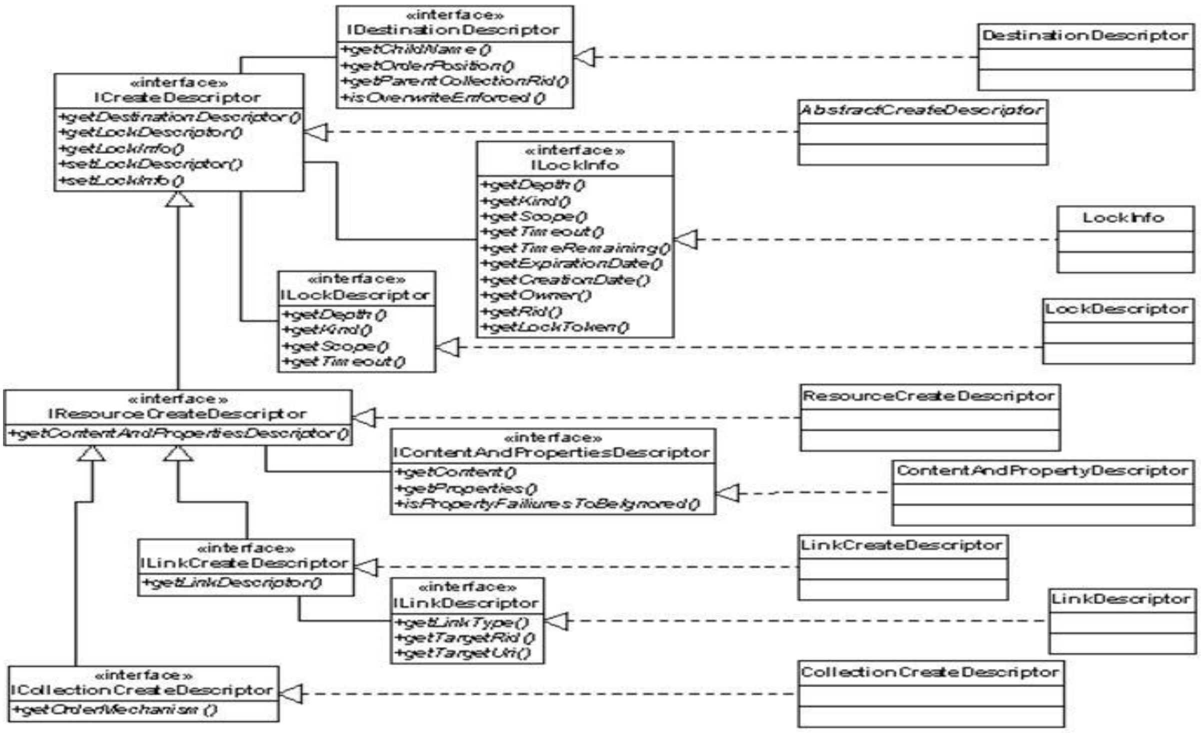
```
public class ZipfileNamespaceManager extends AbstractSubManager implements IMutableNamespaceManager
{
```

This introduced several new method signatures to be implemented, depending on the supported options. It was not always mandatory to implement all mutable method signatures. Those that the developer didnt want to be implemented was thrown an OperationNotSupportedException.

In this example, the creation and deletion of resources was provided, so one had to implement the createResource() and deleteResource() method.

```
public IResourceHandle createResource(ICreateDescriptor d) throws ResourceException;
public List deleteResource(IResourceHandle handle) throws ResourceException,
OperationNotCompletedException;
```

Looking deeper into the createResource() method, the developer had to know the concept of create descriptors. Create descriptors contain all the information required for creating a certain resource object in the connected backend system.



In this implementation the developer considered the IResourceCreateDescriptor for resource (document) creation and the ICollectionCreateDescriptor for collection (folder) creation. Both could be supported by the backend system.

```
if ((d instanceof IResourceCreateDescriptor) || (d instanceof ICollectionCreateDescriptor)) {
```

The CreateDescriptor offers the possibility to get the RID of the collection, the resource is created in, and the name of the resource which was passed from the RF to the Repository Manager.

```
IRid parent = d.getDestinationDescriptor().getParentCollectionRid();
String name = d.getDestinationDescriptor().getChildName();
```

Names were normally passed in an encoded way by the client, so the developer had to decode them first before he could pass it to the backend system.

```
name = UriCodec.Decode(name);
```

To create the resource in the corresponding parent collection, the developer had to lookup the handle for the parent RID. In this example the lookup() method of the Repository Manager class is used.

```
ZipfileHandle handle =
   ((ZipfileHandle)this.repositoryManager.lookup(parent));
```

Resources can only be created on collections, therefore we have to check the collection flag in the backend implementation. If the current handle's node is not a collection you should throw a Resource Not Collection Exception.

```
Node node = handle.getNode();
if (!node.is Collection()) {
   throw new Resource Not Collection Exception(parent,
       "parent is not a collection!");
}
FolderNode pn = (Folder Node)node;
Node child = isCol ? new Folder Node(name, pn) : new Node(name, pn);
```

After the resource was created as a Node reference to the backend system, the developer had to get the content and properties out of the create descriptor and set them to the node object. The properties should be retrieved out of the specialized IResourceCreateDescriptor or ICollectionCreateDescriptor since there are collection and resource specific system properties already set.

```
// create content
IMutable Content content = cpd.getContent();
child.setContent(
   (content==null ? null : content.getInputStream())
);
```

```
// create properties
List pl = cpd.getProperties();
child.setProperties(pl);
```

If the creation was commited to the backend successfully, the developer had to return a valid new handle of the created resource.

```
return new ZipfileHandle(child);
```

In case of a failure, you should return null.

The deleteResource() method is quite simple. The handle to be deleted is passed by the RF. You have to take care that all information (content, properties, children elements) related to the handle has been cleaned in the backend.

```
public List deleteResource(IResourceHandle handle)
   throws ResourceException, OperationNotCompletedException {
      List l = new ArrayList();
      if (handle instanceof ZipfileHandle) {
         ((ZipfileHandle)handle).getNode().delete();
         l.add(handle.getRid());
      }
   return l;
}
```

This method must return a list of RIDs of all deleted handles. By default the list only contains one RID, the one of the deleted handle passed into this method. In case of deleting a collection you may also provide the RIDs of the children handles, which are deleted also.

If one does not specify a supported option for copy and move, the framework automatically project the move operation to a create and delete call, the copy operation is projected to a create call. In this case the developer had not to implement the methods but had to throw an OperationNotSupportedException.

```
public List copyResource(IResourceHandle
handle, ICopyDescriptor cd)  throws ResourceException,
OperationNotCompletedException {
   throw new OperationNotSupportedException(handle.getRid(), "copyResource() not supported.");
}
public List move Resource(IResourceHandle
handle, IMove Descriptor md)  throws ResourceException,
OperationNotCompletedException {
   throw new OperationNotSupportedException(handle.getRid(), "moveResource() not supported.");
}
```

Content Manager Aspect

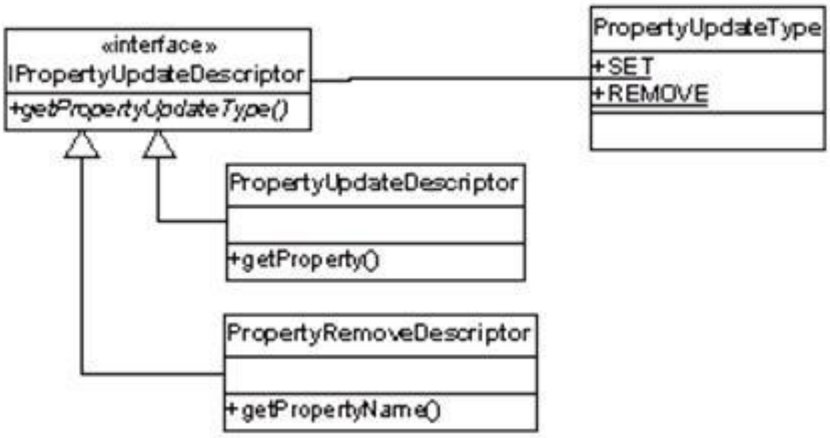
The mutable content manager only deals with updating the content in the backend system. He will set the mutable content object to the handle provided by the RF.

```
public void setContent(IResourceHandle handle,
IMutableContent content,  boolean compareETags) throws
ResourceException, ContentMetadataMismatchException {
   if (handle instanceof ZipfileHandle) {
      Node n = ((ZipfileHandle)handle).getNode();
      if (!n.isCollection()) {
         n.setContent(content.getInputStream());
      }
   }
}
```

In this implementation, the content only gets set if the handle is not a collection. In general, if the backend system allows collections to contain content, the content could also be set for collections. Keep in mind that, for this case, the standard KM application does not provide a client side API or a user interface.

Property Manager Aspect

Similar to the namespace manager aspect, the property manager aspect provides the concept of descriptors. The IPropertyUpdateDescriptor was passed by the RF to the property manager implementation.



Like the immutable version of the property manager, the mutable version had several method signatures to be implemented for single and mass calls.

```
public void updateProperty(IResourceHandle handle, IPropertyUpdateDescriptor pud)   throws
ResourceException;
```

In this example, most of the methods are simply delegated to one implementation. In a productive Repository Manager implementation, it could be better to have all methods implemented for better performance.

```
public void updateProperties(IResourceHandle handle, List puds) throws ResourceException {
   IPropertyUpdateDescriptor pud = (IPropertyUpdateDescriptor) puds.get(i);
```

Both updating/setting and deletion is done in the updateProperties() method. The operation to be performed had to be determined by the passed IPropertyUpdateDescriptors.

```
// set or update a property
if (pud instanceof PropertySetDescriptor) {
   IProperty p = ((PropertySetDescriptor)pud).getProperty();
   Iterator iter = props.iterator();
   IProperty pr = null;
   while (iter.hasNext()) {
      pr = (IProperty) iter.next();
      if (pr.getPropertyName().equals(p.getPropertyName()))
         iter.remove();
   }
   props.add(p);
}
// remove a property
if (pud instanceof PropertyRemoveDescriptor) {
   IPropertyName p = ((PropertyRemoveDescriptor)pud).getPropertyName();
   Iterator iter = props.iterator();
   IProperty pr = null;
   while (iter.hasNext()) {
      pr = (IProperty) iter.next();
      if (pr.getPropertyName().equals(p))
         iter.remove();
   }
}
n.setProperties(props);
```

Using the Security Checker mechanism

In most Repository Manager implementations it is useful to protect the repository’s data from unauthorized access. Therefore you have the possibility either to develop your own security aspect or use the standard ACL security manager to handle the permissions on resource access level.

An own security aspect only makes sense if the connected backend system exposes security entities to be wrapped by an own security manager. The security entities can be visualized through KM’s details iView permission dialogs.

To make a Repository Manager enabled for using the standard ACL security mechanism, you simply adopt some coding parts in the different Repository Manager aspects.

The security checks are relevant for all aspects and have to be handled separately in each aspect's implementation. A single security check should be done before the method is executed. The security check was executed through the SecurityChecker class which encapsulated all standard security checks for the aspects. The following list shows which security check method had to be called in which aspect's methods.

SecurityChecker method	Unified aspect	Where to implement?
checkReadProperties(IResourceHandle)	Namespace	getCollectionOrderMechanism()
	Namespace	getLinkDescriptor()
	Property	getProperty()
	Property	getProperties()
	Property	getListedProperties()
	Property	getAllProperties()
checkReadContent(IResourceHandle)	Content	getContent()
checkListChildren(IResourceHandle)	Namespace	findResources()
checkCreateChild(IResourceHandle)	Namespace	createResource()
	Namespace	copyResource() [only destination collection]
	Namespace	moveResource() [only destination collection]
checkRemoveChild(IResourceHandle)	Namespace	moveResource() [only source parent collection]
checkRename(IResourceHandle)	Namespace	moveResource()
checkModifyProperties(IResourceHandle)	Namespace	reorderCollection()
	Namespace	setCollectionOrderMechanism()
	Namespace	setLinkDescriptor()
	Property	updateProperty()
	Property	updateProperties()
checkModifyContent(IResourceHandle)	Content	setContent()
checkModifyAll(IResourceHandle)	Versioning	checkIn()
	Versioning	checkOutInPlace()
	Versioning	undoCheckout()
	Namespace	copyResource()
	Namespace	setVersionControlEnabled()
checkDelete(IResourceHandle)	Namespace	copyResource()
	Namespace	moveResource()
checkDelete(IResourceHandle parent, IResourceHandle resource)	Namespace	deleteResource()
checkCopy(IResourceHandle)	Namespace	copyResource()
checkLock()	Locking	lock()
	Locking	unlock()
	Locking	refreshLock()

Each aspect provides access to the SecurityChecker class by calling the protected method getSecurityChecker():

```
this.getSecurityChecker().checkListChildren(<IResourceHandle>);
```

If permission is denied, the method throws an AccessDeniedException which can be caught, or re-throws in the corresponding method.

Keep in mind that the security checker could be called as infrequently as possible to prevent major performance problems.

Example: Security Checks used in the Namespace Manager Aspect

In the namespace manager aspect, the security checker methods had to be called in almost all important methods.

```
public List findResources(...) throws ResourceException, OperationNotSupportedException {
   this.getSecurityChecker().checkListChildren(handle);
```

The createResource() method also need to be covered with security checks.

```
public IResourceHandle createResource(ICreateDescriptor d) throws ResourceException {
   IRid parent = d.getDestinationDescriptor().getParentCollectionRid();
   String name = d.getDestinationDescriptor().getChildName();
   ZipfileHandle handle = ((ZipfileHandle)this.repositoryManager.lookup(parent));
   this.getSecurityChecker().checkCreateChild(handle, false);
```

The security checks had to be done in a later state of the method call. Unless the developer had no handle passed through the RF, he had to lookup the required parent in a first step.

Configuration

Read/write modes

There were no special settings to the configuration files of the Repository Manager instance since the developer had created a valid one for the immutable version.

```
<property name="class" value="com.sap.bc.rf.manager.ZipfileRepositoryManager" />
<property name="namespacemgr. class" value="com.sap.bc.rf.manager.ZipfileNamespaceManager" />
<property name="contentmgr.class" value="com.sap.bc.rf.manager.ZipfileContentManager" />
<property name="propertymgr.class" value="com.sap.bc.rf.manager.ZipfilePropertyManager" />
<property name="lockmgr.class" value="com.sap.bc.rf.manager.ZipfileLockManager" />
```

Security Checker

To use the features of the SecurityChecker the developer had to maintain the standard ACLSecurityManager entry in the *.co,xml configuration file (see [REP], Configuring & Deployment). The SecurityManager class for the standard AclSecurityManager had to be specified by the securitymgr.ref attribute with the value AclSecurityManagerMI:

```
<property name="securitymgr.ref" value="AclSecurityManagerMI" />
```

Changes in Repository Manager Configuration

Since the introduction of the security checker, the configuration file (*.cc.xml) of the Repository Manager had to be changed to the following new format:

```
<ConfigClass name="com.sap.bc.rf.manager.ZipfileRepositoryManager"
     extends="NewRepositoryManager" hotReload="false" hotLoad="false" hotUnload="false">
   <attribute name="name" type="string"/>
   <attribute name="prefix" type="string" default="/zip"/>
   <attribute name="class" type="class" constant="com.sap.bc.rf.manager.ZipfileRepositoryManager"/>
   <attribute name="archive" type="string" default="D:/rm.zip"/>
</ConfigClass>
```

The extended ConfigClass was derived from NewRepositoryManager.

5.2 Conclusions

Based on the objectives and results:

There was a created, and integrated Content-Incentive-Usability Framework for Corporate Portal Design for Unga Group Limited. By increasing the use of the Internet, Unga Group Limited had tried to have a more integrated framework system within their work stations. With the development of the framework and portal, Unga Group Limited can transform more effectively and offer more varied services to the consuming community.

5.3 Recommendations

Unga Group Limited should invest more in content management tools that offer abilities to integrate, classify, and codify knowledge from various sources.

Unga Group Limited should invest more in Knowledge sharing tools that support sharing knowledge between people or other agents. And in knowledge search and retrieval systems that enable search and retrieval and have some Unga Group Limited should be involved in more knowledge discovery abilities and general KMS which is s system that propose an overall solution for a company's knowledge management needs.

Unga Group Limited should acquire the ability to present a unified view of corporate information that integrates information from different organizational repositories instead of having corporate information spread across many sources within the organization.

Unga Group Limited should acquire the ability to create a shared community because they present a natural forum for online collaboration by assembling a set of content and services to which members of a group have special accesses. Collaboration can also entail offering native portal services such as threaded conversation, project management tools such as task lists, calendars, document sharing or instant messaging.

REFERENCES

[1] Amidon, D. M., & Macnamara, D. (2009). The 7 C's of knowledge leadership: innovating our future. In C. W. Holsapple (Ed.), *Handbook on Knowledge Management* (Vol. 1, pp. 539-552). Berlin/Heidelberg: Springer-Verlag.

[2] Bose, R. (2010). Knowledge management capabilities & infrastructure for e-commerce. *Journal of Computer Information Systems, 42*(5), 40-49.

[3] Chou, T., Hsu, L., Yeh, Y., & Ho, C. (2011). Towards a framework of the performance evaluation of SMEs' industry portals. *Industrial Management & Data Systems, 105*(4), 527-544.

[4] Delio, M. (2008). Keys to collaboration: 7 bottom-line strategies to promote knowledge sharing. *Knowledge Management, 1*(1).

[5] Department.of.Navy. (2011). Knowledge centric organization (KCO) assessment. Retrieved June, 2002, from http://www.fgipc.org/02_Federal_CIO_Council/Downloads/90_KCOSurvey.doc

[6] Dorfman, P. (2011). The accidental knowledge manager. *Knowledge Management, 4*(2), 36-41.

[7] Eraut, M. (2012). Non-formal Learning and Tacit Knowledge in Professional Work. *British Journal of Educational Psychology, 70*, 113-136.

[8] Fernandez, M., Florescu, D., Levy, A., & Suciu, D. (2012). Declarative Specification of Web Sites with Strudel. *VLDB Journal, 9*(1), 38-55.

[9] Gaines, B. (2009). Organizational knowledge acquisition. In C. W. Holsapple (Ed.), *Handbook on Knowledge Management* (Vol. 1, pp. 317-348). Berlin/Heidelberg: Springer-Verlag.

[10] Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., & Widom, J. (2011). *Integrating and Accessing Heterogeneous Information Sources in TSIMMIS.* Paper presented at the AAAI Symposium on Information Gathering, Stanford, California.

[11] Gill, P. J. (2011). Once upon an enterprise. *Knowledge Management, 4*(5), 24.

[12] Glavinic, V. (2010). *Introducing XML to integrate Applications within a Company.* Paper presented at the 24th International Conference on Information Technology Interfaces, Zagreb, Croatia.

[13] Grant, R. M., & Baden-Fuller, C. (2011). *A knowledge-based theory of inter-firm collaboration.* Paper presented at the Academy of Management Best Papers.

[14] Hanley, S., & Malafsky, G. (2009). A guide for measuring the value of KM investments. In C. W. Holsapple (Ed.), *Handbook on Knowledge Management* (Vol. 2, pp. 369-394). Berlin/Heidelberg: Springer-Verlag.

[15] Heijden, H. v. d. (2009). Factors influencing the usage of websites: the case of a generic portal in The Netherlands. *Information & Management, 40*, 541-549.

[16] Holsapple, C. W., & Jones, K. G. (2013). Exploring primary activities of the knowledge chain. *Knowledge and Process Management, 11*(3), 155-174.

[17] Holsapple, C. W., & Jones, K. G. (2011). Exploring secondary activities of the knowledge chain. *Knowledge and Process Management, 12*(1), 3-31.

[18] Jayatilaka, B., Schwarz, A., & Hirschheim, R. (2009). Determinants of ASP Choice: An Integrated Perspective. *European Journal of Information Systems, 12*(3), 210-224.

[19] Jin, Y., Decker, S., & Wiederhold, G. (2011). *OntoWebber: Model-Driven Ontology-Based Web Site Management.* Paper presented at the The 1st International Semantic Web Working Symposium (SWWS'01), Stanford, USA.

[20] Kim, Y. J., Chaudhury, A., & Rao, H. R. (2010). A Knowledge Management Perspective to Evaluation of Enterprise Information Portals. *Knowledge and Process Management, 9*(2), 57-71.

[21] King, W. R. (2006). Knowledge Sharing. In D. G. Schwartz (Ed.), *Encyclopedia of Knowledge Management* (pp. 493-498). Hershey: Idea Group Reference.

[22] Lai, H., & Chu, T.-H. (2010). Knowledge management: a review of industrial cases. *Journal of Computer Information Systems, 42*(5), 26-39.

[23] Lapre`, M. A., & Van Wassenhove, L. N. (2011). Creating and transferring knowledge for productivity improvement in factories. *Management Science, 47*(10), 1311-1325.

[24] Liebowitz, J., & Chen, Y. (2009). Knowledge sharing proficiencies. In C. W. Holsapple (Ed.), *Handbook on Knowledge Management* (Vol. 1, pp. 409-424). Berlin/Heidelberg: Springer-Verlag.

[25] Lin, C., Wu, S., & Tsai, R. (2011). Integrating perceived playfulness into expectation-confirmation model for web portal context. *Information & Management, 42*, 683-693.

[26] Markus, M. L. (2011). Toward a theory of knowledge reuse: types of knowledge reuse situations and factors in reuse success. *Journal of Management Information Systems, 18*(1), 57-93.

[27] Marwick, A. (2011). Knowledge Management Technology. *IBM Systems Journal, 40*(4), 814-830.

[28] Massey, A. P., Montoya-Weiss, M. M., & O'Driscoll, T. M. (2010). Knowledge management in pursuit of performance: insights from Nortel Networks. *MIS Quarterly, 26*(3), 269-289.

[29] Mullich, J. (2011). Growing a knowledge management system. *Knowledge Management, 3*(3), 54.

[30] Nonaka, I., & Takeuchi, H. (2011). *The Knowledge Creating Company*. Oxford: Oxford University Press.

[31] O'Dell, C. (2010). *Stages of implementation: a guide for your journey to knowledge management best practices*. Houston: American Productivity & Quality Center.

[32] O'Dell, C., Elliott, S., & Hubert, C. (2009). Achieving knowledge management outcomes. In C. W. Holsapple (Ed.), *Handbook on Knowledge Management* (Vol. 2, pp. 253-288). Berlin/Heidelberg: Springer-Verlag.

[33] Shneiderman, B. (2008). *Designing the User Interface* (Third ed.): Addison Wesley Longman.

[34] Smith, H. A., & McKeen, J. D. (2009). Knowledge management in organizations: the state of current practice. In C. W. Holsapple (Ed.), *Handbook on Knowledge Management* (Vol. 2, pp. 395-410). Berlin/Heidelberg: Springer-Verlag.

[35] Van Alstyne, M. W. (2011). Create colleagues, not competitors. *Harvard Business Review, 83*(9), 24-28.

[36] Yang, Z., Cai, S., Zhou, Z., & Zhou, N. (2011). Development and validation of an instrument to measure user perceived service quality of information presenting Web portals. *Information & Management, 42*, 575-589.