2009-06-16

A Functional Framework for Content Management

Robert Emer Broadbent
Brigham Young University - Provo

A FUNCTIONAL FRAMEWORK FOR

CONTENT MANAGEMENT

by

Robert E. Broadbent

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

School of Technology

Brigham Young University

August 2009

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Robert E. Broadbent

This dissertation/thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Joseph J. Ekstrom, Chair
Date	Michael G. Bailey
Date	Bret R. Swan

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the dissertation/thesis of Robert E. Broadbent in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Joseph J. Ekstrom Advisor
Chair, Graduate Committee

Accepted for the Department

Ronald E. Terry
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering
and Technology

ABSTRACT

A FUNCTIONAL FRAMEWORK FOR CONTENT MANAGEMENT

Robert E. Broadbent

School of Technology

Master of Science

This thesis proposes a functional framework for content management. This framework provides concepts and vocabulary for analysis and description of content management systems. The framework is derived from an analysis of eight content management systems. It describes forty-five conceptual functions organized into five functional groups. The functionality derived from the analysis of the content management systems is described using the vocabulary provided by the functional framework. Coverage of the concepts in the existing systems is verified. The utility of the framework is validated through the creation of a prototype that implements sufficient functionality to support a set of specific use cases.

ACKNOWLEDGEMENTS

I am extremely grateful for the support and encouragement I have received during the writing of this thesis. I would like to thank my graduate chair, Dr. Joseph Ekstrom, for his mentorship and the amazing opportunities he has provided me throughout my academic career at Brigham Young University. I would like to thank my graduate committee, Dr. Michael Bailey and Dr. Bret Swan, for their contributions and time. I am forever grateful to my wife, Eve, for her endless support and encouragement throughout this entire process. Finally, I would like to thank my friends at Cemaphore Systems for their financial support and the chance to turn academic ideas into reality.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

This thesis proposes a functional framework for content management. This framework provides concepts and vocabulary for analysis and description of content management systems. The framework is derived from an analysis of eight content management systems. It describes forty-five conceptual functions organized into five functional groups.

Conceptual functions represent generalized CMS functionality such as form input, document conversion, and Web content publication. Conceptual functions are organized into the general functional groupings of content capture, content delivery, content management, administration, and system architecture.

A functional module implements specific functionality, such as the conversion of a word processing document to a PDF-formatted document. A conceptual function is an abstract representation of functionality, which can be implemented by any number of functional modules. In practice, a functional module is an implementation of an application program interface (API).

The functionality observed in the analysis of the content management systems is later described using the vocabulary provided by the functional framework. A table was assembled, which represented the conceptual functions found in the analysis of the CMSs. This table was refined until the conceptual functions could not be easily decomposed and redundancy was removed. Completeness was verified through demonstrating coverage of the existing system functionality using only the conceptual functions available from the refined table.

The utility of the framework is validated through the creation of a prototype that implements sufficient functionality to support a set of specific use cases.

1.1 Background

During the author's undergraduate and graduate studies, he has had the opportunity to research and experiment with open-source text-indexing software libraries. These libraries provide the functionality to quickly locate the most relevant documents for a given meta-data query. The documents one might search for range from word processing documents, web pages, and email to images and video. The commonality among frequently searched for documents is that they contain unstructured data. This data is referred to as content.

The collection of information represented by content is often one of the most valuable assets belonging to an organization or an individual. Content may be the result of countless hours of work or it may be an irreplaceable reminder. Content assets may include multi-media for a new advertising campaign or the documentation for a company's proprietary processes or the daily communication between employees. Content may be of personal importance and sentimental value. An individual's collection of content may include the photos and videos of family and other loved ones. It may be one's personal music collection or the word processing documents for an important assignment.

The goal regarding content is to manage and use it effectively and efficiently. Content-metadata indexing tools provide the functionality to quickly identify documents whose metadata meet a certain criteria. This allows the content to be used in a time-efficient manner, and has been demonstrated in popular applications such as searching the World Wide Web. However, indexing tools are just a piece of an overall content management infrastructure. Many other functional pieces should be considered to make the best use of content.

To gain a more comprehensive view of content management tools, one must look to content management systems (CMSs). CMSs provide a collection of functionality for working with content. This functionality includes content capture, content storage, and content publication, as well as the before-mentioned metadata indexing. CMSs, are often purpose specific or content-type specific, yet they still provide a wide range of functionality.

In order to make the best use of content, one should have an understanding of CMSs. Unfortunately, gaining this understanding can be a difficult task. The field of content management is rapidly evolving due to the considerable efforts invested into the research and development of CMSs. There are hundreds of content management products, which approach content management in many different ways. Inconsistent and overlapping terminology promoted by vendor marketing adds to the confusion. To combat this confusion, models or frameworks are created to provide a context within which to understand a given subject.

1.2 Problem Statement

A conceptual framework is needed to provide a practical foundation for CMS research and development. Inconsistent terminology and overlapping functionality are causing confusion regarding CMSs. This is due to the lack of validated CMS frameworks at the practical level, which specify CMS component types and functional components. A valid conceptual framework is especially important to address considering the proliferation in CMS technologies and tools. There is a growing disconnect between existing CMS models and tools, which cause problems in identifying and implementing CMS functionality.

1.3 Thesis Statement

This research proposes a functional framework, which provides the concepts necessary to describe use cases from an operational perspective. The framework is validated through the creation and implementation of a prototype containing functional modules derived from the conceptual functions. The prototype implements a set of use cases described in the delimitations section of chapter 1.

1.4 Research Methodology Overview

This thesis identifies the common CMS component types and their conceptual functions across a breadth of popular CMS technologies for which documentation of functionality was available. A study and evaluation of existing CMS conceptual frameworks was performed to determine their practicality at identifying and implementing CMSs. These frameworks did not address the practical aspects needed to describe the uses and implement content management. The lack of existing functional frameworks to describe CMS functional modules at a practical level indicated a need to develop a new conceptual framework and attempt to validate it. This research analyzed eight CMSs that covered a breadth of functionality. Based on this analysis and a review of existing frameworks, a functionally-oriented conceptual framework is proposed. The framework represents the conceptual functions as found in Appendix A. The new framework is validated by implementing a prototype with associated API that demonstrates the viability of the concepts.

1.5 Assumptions

This thesis assumes that a conceptual framework for content management can be identified from an analysis of eight content management systems. It also assumes that an adequate variety of types of content management systems are surveyed.

It is assumed that the prototype's use cases provide an adequate validation and demonstrated use of the framework. These use cases are presented in the following delimitations section.

1.6 Delimitations

This thesis provides prototype and API of a content management system based on the framework for content management. The implementation is a system for the management of documents based on a hierarchically-defined set of document types.

The following use cases are supported by the implementation:

- Manage hierarchically-defined document classes with three possible attribute types
- Capture document metadata based on document classification
- Capture Microsoft Word 2003, Microsoft Word 2007, and PDF documents via Web-based document upload
- Manage individual document lifecycles
- Support post-capture document processing of the following set of actions:
    - Detect the captured document content type by document filename extension
    - Convert the captured document to universal-format document type, which is the PDF file format
    - Convert the captured document to a plain-text document
    - Create preview images of the first ten pages of the captured document
    - Create a thumbnail image of first page of the captured document
    - Index captured document text
    - Index the captured document metadata
- Identify the documents whose document text and/or metadata match a given search criteria
- Support workflow for approval of document publication
- Publish the document and document metadata by Web Page via the World Wide Web
- Publish the document by document download via the World Wide Web

1.7 Thesis Chapters Overview

Chapter 2 of this thesis is a review of literature. It provides a background to the subject of CMSs and an overview of the variety of CMS products that are currently available. The chapter also provides an overview of some existing conceptual frameworks for content management, which fail to meet the practical concerns that the new conceptual framework is intended to address. Finally, this chapter provides three example case studies of adaptations of CMSs in which the conceptual frameworks were either unavailable or did not address the practical issues, which the new framework addresses.

Chapter 3 is a report on the component functionality identified in the eight CMSs chosen for analysis. This chapter includes concept maps, which present the functionality of each CMS in a hierarchically-organized fashion. A brief overview of each CMS and its functionality is also provided in chapter 3.

Chapter 4 describes the synthesis of the component functionality found in chapter 3 and the organization of the component functionality into functional types. This chapter then presents a new conceptual framework for content management functionality.

Chapter 5 discusses the implementation architecture and API derived from the use cases presented in chapter 1 and the implementation of the architecture used to validate the framework. This chapter identifies the functional components from the new framework, which were implemented, and provides an overview of the API used to exercise the implemented functionality.

Chapter 6 presents the conclusions of this thesis. This discussion includes a review of the successful validation of the new framework by and recommendations for future research based on framework.

2 Review of Literature

2.1 Early Content Management Systems

There have been ongoing efforts to manage information and make it available in usable ways. One of the early proposals for a tool that could make information more accessible and usable was the “memex,” whose ideas inspired innovation and are reflected in many of today’s CMSs.

In July of 1945, Dr. Vannevar Bush published an article entitled “As We May Think” in *The Monthly Atlantic*. “As We May Think” was Dr. Bush’s admonition to scientists to turn their intellectual efforts from creating the tools of war to the goal of making accessible the knowledge of mankind. At the time his article was published, Dr. Bush was the Director of the United States Office of Scientific Research. He hoped to positively influence the efforts of the approximately six-thousand American scientists he was tasked with directing (Bush 1945).

In his article, Dr. Bush described his vision for the investigation and command of human knowledge. Part of this vision was the “memex”, a mechanized library of books, pictures, newspaper, microfilm, and other print materials. The memex would mechanically fetch the desired materials and display them on large screens. Keyboards and speech recognition technology would be employed to allow the user to quickly and naturally navigate and annotate information (Bush 1945).

One feature of the memex was the ability to associate related pieces of information. One document could be associated with another and create what Dr. Bush called a trail. Trails of information were persistent and easily navigable, thus allowing one to quickly explore a vast maze of information (Bush 1945).

Decades later, Dr. Ted Nelson found inspiration in the idea of the memex and the memex's trails of related information. Dr. Nelson began advocating the implementation of the concept of trails using computer technology. He proposed making trails applicable to specific text rather than an entire document and envisioned a global network of computers containing documents. In 1965, Dr. Nelson proposed the term "hyperlinking" to identify his version of electronic trails (Wolf 1995).

2.2 Today's Content Management Systems

Today, the World Wide Web is enabling Dr. Bush's vision of making the knowledge of mankind accessible. The Word Wide Web is similar to a memex composed of the contents of the millions of Web servers, and the World Wide Web provides a vast library of documents and digital media, which are, in most cases, interconnected using hyperlinks.

However, the World Wide Web and other information repositories still face the challenges of the explosive growth of information. The collection of digital information is constantly growing and ever increasing demands are being placed on how this information is managed and delivered. In many cases, these demands are being met by tools called content management systems (CMSs).

Content management is a sub-component of the field of information management. Information management deals with the management of all content within an organization, including structured and unstructured data. Structured content is strongly typed data, where pieces of data have identifiable meaning or purpose. Unstructured data lacks a precise structure and cannot be simply interpreted (Terra 2002).

The tools for managing structured information are called database management systems. The tools for managing unstructured information are called content management systems (Reimer 2002, 18-19).

The volume of unstructured content produced by organizations is now growing faster than the volume of structured content. The growth of unstructured content combined with the increasing legal and industrial regulations regarding the treatment of digital information has made the management of unstructured content a field of increasing research and development efforts (Reimer 2002, 17).

The increase in unstructured content in organizations is estimated to be growing at a rate of 800 MB per person per year (Gingell 2005, 5). This increase in digital content is driving the emergence of many different content management system platforms, which support a variety of content-management-oriented applications (Tramullas 2005).

The content management process consists of the creation, approval, delivery and management of unstructured content. Content creation is the process of content owners creating and editing their content. The content approval process is based on a predefined workflow process that allows a content manager to approve content prior to publication. Content delivery is the process for content publication. Finally, content management is the organization and tagging of content with metadata so that it is easily found. More fully featured products include the ability to perform automatic metadata tagging, content version control, and expiration of content (McNay 2002, 298-399).

2.3 Diversity in Today's CMS Market

A typical CMS provides many features, including the following: document publishing, workflow, information repository processes; an information repository; tools for integrating external information sources; and template capabilities (Tramullas 2005).

CMSs have become increasingly complex and sophisticated systems. This complexity has led to the specialization of CMSs (Tramullas 2005). The following provides an enumeration of some of the different types of CMSs.

2.3.1 Content Management Platform

Content management platform consists of an environment and development tools on which content management solutions can be implemented. Examples of content management platforms include Zope (www.zope.com), Typo3 (typo3.com), Midgard Project (www.midgard-project.org), OpenCMS (www.opencms.org/en), and Apache Lenya (www.opencms.org/en). (See Tramullas (2005) for more information).

2.3.2 Content Portals

Content portals manage and administer content and services as web information services. Services include news publications, forums, surveys, content syndication, profile and group management, personalization of information and presentations, etc.

Content portals are typically designed around a modular architecture. Examples of content portals include PHP Nuke (php-nuke.org), Drupal (drupal.org), Mambo (mambo-foundation.org), and Plone (plone.org). (See Tramullas (2005) for more information).

2.3.3 Virtual Classrooms

Virtual classroom management systems support the publication of content for online learning and collaboration through forums, chat, on-line evaluation, etc. Examples of open-source virtual classroom systems include Claroline (claroline.org) and Moodle (moodle.org). (See Tramullas (2005) for more information).

2.3.4 Digital Libraries

Digital Library systems organize content around users, collections, and services. These systems often provide tools and management and collaboration services organized around collections. Examples of digital library systems include Fedora (www.fedora.info), DSpace (www.dspace.org), and Greenstone (www.greenstone.org). (See Tramullas (2005) for more information).

2.3.5 Digital Publications

Digital Publication systems are focused on digital publications such as newspapers and magazines. These systems provide functionality for managing content editing, creation, and publication. Digital publication systems include Cofax (www.cofax.org), Open Journal Systems (pkp.sfu.ca/?q=ojs), and ePrints (www.eprints.org). (See Tramullas (2005) for more information).

2.3.6 Collaboration Systems

Collaboration systems provide tools for working in a group such as the support of groups of users working jointly on projects. These types of systems manage workflow, users, process and workflow control points, content deliverables, and these systems provide collaboration tools for communication and controlling activities. Collaboration systems include the concept of wikis, which enable the "elaboration of documents for interest-sharing communities". Example collaboration systems include eGroupware (www.egroupware.org), phpCollab (www.php-collab.com), and wiki systems, such as MediaWiki (www.mediawiki.org). (See Tramullas (2005) for more information).

The many features of a CMS are best utilized in collaborative and distributed work environments (Tramullas 2005). Dynamic, up-to-date content can be encouraged by giving authors ownership of their content. The decentralized process of content management by multiple authors and groups is referred to as distributed authorship or distributed content management. This encourages better collaboration within groups and more relevant, specialized content (McNay 2002, 397).

2.3.7 Weblogs

Weblogs are usually single user and simple workflow publication systems, which enable non-technical users to publish documents on the World Wide Web. WordPress (wordpress.org) is an example of an open-source blog system (Tramullas 2005).

2.3.8 Web Content Management

Web content management is "the creation, publishing and management of company information and documents on the web" (McNay 2002, 397). Over the last 10 years, publishing to the web has evolved in complexity. Web publication methods and techniques, as well as products have evolved in response to this complexity (Tramullas 2005).

Web content management (WCM) is important to the creation of dynamic and useful websites. Content management for the web entails the organization of web content in a logical, consist, and navigable way. (McNay 2002, 396) Aspects of distributed authorship and currency of information are important aspects of web publication. (McNay 2002, 397).

Presentation of content in WCM can take an XML-based approach using XSL style sheets, which allow content and layout to be dynamically generated for the target environment. Pieces of content are assembled into documents and styled as needed. This approach supports the standardization of the look of a website (Weitzman 2004, 71). The process of making prepared content available on the web is known as delivery management (McNay 2002, 397).

2.3.9 Document Management

Document management software manages document lifecycles. This includes authoring, collaborative authoring, and archival. Other features are indexing, check-in and check-out of content (also called library services), versioning, annotations, workflow and life-cycle management (Moore 2002, 6).

2.3.10 Integrated Document Management

Integrated document management software scans, indexes, retrieves and archives digital images. Content may include images of “text, graphics, engineering drawings and photographs”. These systems include workflow and some document management functionality (Moore 2002, 7).

2.3.11 Digital Asset Management

Digital asset management software manages the lifecycle of digital content such as images. DAM includes Media Asset Management (MAM) (Moore 2002, 7).

2.3.12 Media Asset Management

Media asset management software handles high complexity digital asset types such as video and sound (Moore 2002, 7-8).

2.3.13 Computer Output to Laser Disk/Enterprise Report Management

Computer-output-to-laser-disk/enterprise-report-management (COLD/ERM) software manages high-volume reports from other systems. These tools include report generation tools and report indexing functionality (Moore 2002, 8).

2.3.14 Records Management

Records management software maintains documents through the long-term document lifecycle. This includes the destruction/deletion of documents at specific points in time (Moore 2002, 7).

2.3.15 Ontology Management

CMSs sometimes keep track of taxonomies and their interrelationships. This structure of information is called an ontology. An ontology provides a definition or description of the various semantics of a given word; words can have different meanings and uses depending on the context in which they are used.

In their 2003 paper, Maedche, et. al, describe a system that provides a service within an ECM that manages ontologies in a way similar to how a document management system manages text fragments. This service performs functions such as normalizing ontologies, performing translations between ontologies, detecting conflicts within an ontology, keeping versions and supporting reverting to an ontology at a specific time, and responding to ontology queries (Maedche 2003).

2.3.16 Enterprise Search

There are inherent difficulties with enterprise search as compared to whole-Internet search. Enterprise search requires a wide range of functional capabilities and is difficult because of the variety of formats involved and the unstructured nature of this data (Mukherjee 2004, 38).

2.3.17 Enterprise Content Management

Enterprise Content Management (ECM) systems provide a complete, integrated solution for content management. The Meta Group defines ECM as "... the technology that provides the means to create/capture, manage/secure, store/retain/destroy, publish/distribute, search, personalize and present/view/print any digital content (i.e. picture/images/text, reports, video, audio, transactional data, catalog, code). These systems primarily focus on the capture, storage, retrieval, and dissemination of digital files for enterprise use" (Munkvold 2006, 71). As the name implies, ECM is also "the category of software that helps you manage all of the unstructured information--or content--in your enterprise." Content can be in many digital forms: "text documents, engineering drawings, XML, still images, audio and video files, and many other file types and formats." ECM allows for integration with desktop applications. ECM can manage and integrate content from a variety of sources (Gingell 2006, 4). It can manage external content from sources such as enterprise resource planning (ERP), customer relationship management (CRM), enterprise portals (Gingell 2006, 4).

ECM is a combination of distinct and interrelated technologies: "enterprise document management (EDM), Web content management (WCM), "digital asset management (DAM), enterprise records management (ERM), business process management (BPM), enterprise content integration (ECI), and collaborative content management (CCM)" (Gingell 2006, 4).

ECM can manage metadata, making content easier to find and search such as categorization schema and tags. ECM supports workflow and lifecycle management, which are "the review, revision, and approval process for any piece of content according to user-defined business rules". ECM can manage and enforce the relationship between pieces of content, and it supports the publication of documents in multiple ways and a variety of formats (Gingell 2006, 4).

2.3.18 A Breadth of Content Management Systems are Needed for an Effective Analsysis

A review of the diversity of CMS products re-emphasizes the tremendous variety of product offerings. These products offer a variety of functionality and use a diverse, often inconsistent, vocabulary. It is important to recognize the variety of product offering when selecting a set of CMS for later analysis in order to represent as much of the available functionality as possible when preparing the new framework.

2.4 Compliance: The Driver of Adoption of Content Management Systems

Compliance is one of the major driving forces in the adoption of content management systems. With significant legal and financial ramifications for a lack of compliance to laws regarding content retention and security, organizations are highly motivated to implement solutions, which will help them meet their regulatory obligations. Compliance regulations vary by industry sector but include (See Gingell (2006) for more information):

- Life sciences: 21 CFR Part 11 (U. S. Food and Drug Administration 2003)
- Financial services: Basal II, SEC 19B-4, SEC 17A-3 and 4 (U.S. Securities and Exchange Commission 2001)

- Oil and gas: OSHA (U.S. Department of Labor: Occupational Safety and Health Administration, n.d.), EPA (U.S. Environmental Protection Agency, n.d.)
- Government: data protection through the Freedom of Information Act (FOIA) (U.S. Department of State Freedom of Information Act, n.d.)
- Public companies: Sarbanes-Oxley Act (U.S. Public Law 107-204, 2002)

2.5 Conceptual Frameworks for Content Management

Conceptual frameworks are important for the understanding, research, use, communication and development of software products. There are several conceptual frameworks that address the subject of content management in various ways. The following is an overview of some important examples.

2.5.1 Structuration Theory Perspective

The structuration theory, as illustrated in Figure 2-1, attempts to identify how processes fit within their social and organizational contexts. Weiseth, et al. describe an environment based on structuration theory which attempts to put content management—collaboration, in particular—into context within the organization (Weiseth 2006).

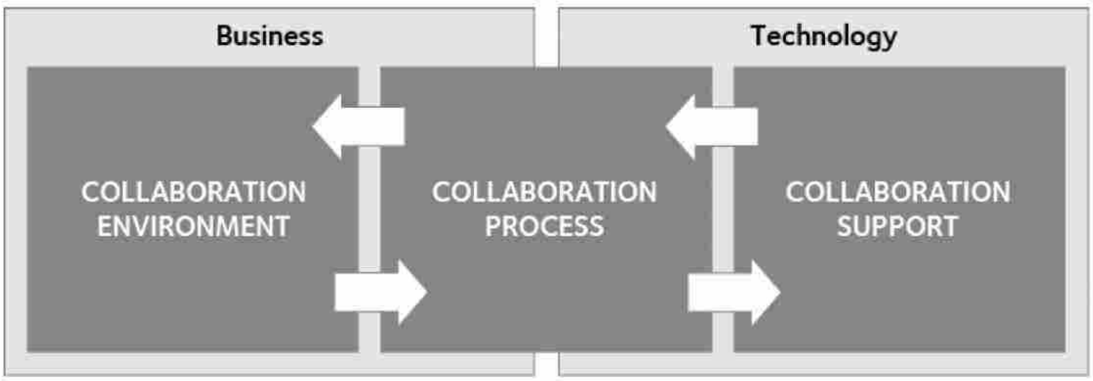


Figure 2-1 The Collaboration Framework (Weiseth 2006)

The structuration theory perspective of collaboration tools includes the collaborative environment, collaboration process and collaboration support within the business and technology contexts (Weiseth 2006).

2.5.2 Research Topics in Enterprise Content Management

Organizational challenges relating to ECM are not only technical. There are also content production and organizational issues, such as the production of new content, which may impact the business processes of the organization (Tyrväinen 2006, 628).

Tyrvaeinen, et. al, presents a framework for ECM research that addresses not only the technological aspects but also the content and organizational contexts (Tyrväinen 2006, 628). This framework includes four perspectives: content, technology, enterprise and process (Tyrväinen 2006, 628).

The content perspective addresses content items, their organization and creation, and use by human and other systems. The content perspective can be further broken down into the following views: the information view, the user view, and the system view (Tyrväinen 2006, 628).

The information view looks at content semantics, representation, and accessibility. This deals with content organization, metadata, and search. The user view deals with the relationship between users and content. This view includes issues with content creation, maintenance, and content use. The system view deals with the system on which content resides, how content is made accessible to users and other systems, content processing and storage, and related standards (Tyrväinen 2006, 628).

The technology perspective addresses research into hardware and software development, standards for content management (Tyrväinen 2006, 628).

The enterprise perspective looks at the environment in which content must be managed. This includes issues relating to the organization, legal, social, and business. (Tyrväinen 2006, 628) This may include the analysis of existing and proposed solutions from the user's perspective, including personalization of content (Tyrväinen 2003, 104).

The process perspective addresses the development and deployment of ECM solutions (Tyrväinen 2006, 628).

Figure 2-2 portrays the four perspectives of the ECM research model.

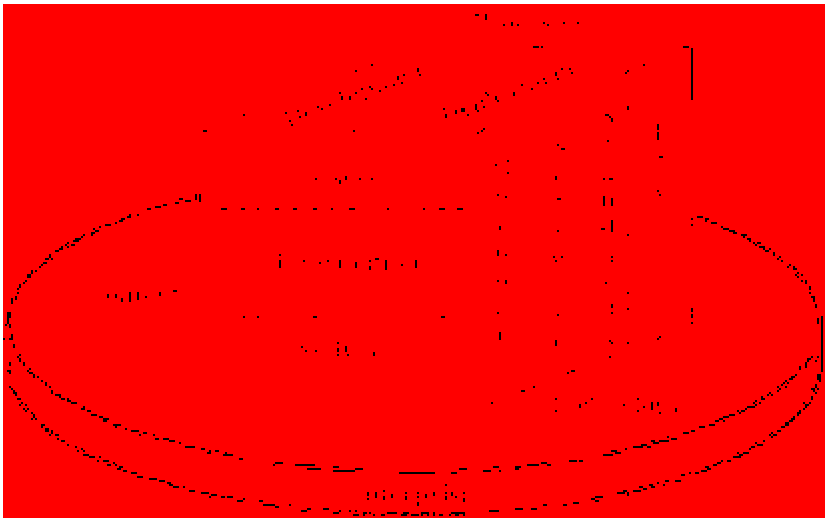


Figure 2-2 A Framework for ECM Research (Tyrväinen 2006, 628)

2.5.3 The Wheel of Collaboration Tools

Weiseth, et al. have identified a framework for collaboration tool capabilities which they call the Wheel of Collaboration Tools (WCT). The Wheel of Collaboration Tools is illustrated in Figure 2-3. This framework supports the specification, analysis and evaluation of collaboration and information management tools. This framework is especially geared towards the identification of the requirements for collaborative tools and aides in "needs analysis, strategy development, feasibility studies, requirements specification and product evaluation." It be used to "assist in reviewing the existing portfolio, analyze the products offered by the vendors, and prepare a basis for a strategy to meet the company's need for collaboration support" (Weiseth 2006).

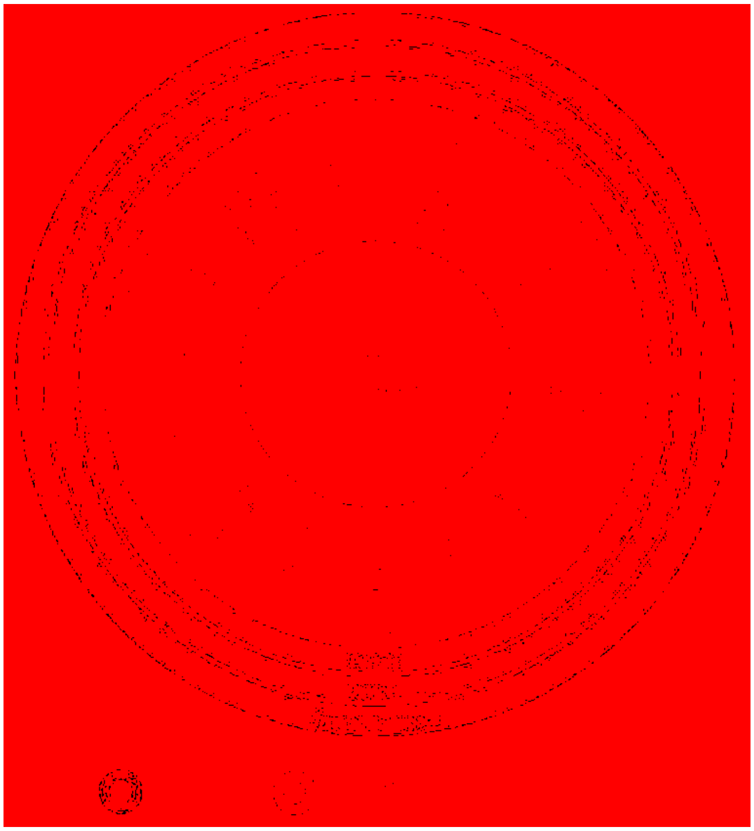


Figure 2-3 The Wheel of Collaboration Tools (Weiseth 2006)

For each of the collaborative processes, sub-processes are identified which have major functional areas. Product analysis and experience with the collaborative process were used to identify processes and sub-processes. The Collaboration functions are broken down into coordination, production, and decision making (Weiseth 2006).

2.5.4 Designing for Entire Lifecycle Support

In his article, Zykov looks at a structured approach for designing content management solutions in enterprises with increasingly voluminous and decentralized data stores (Zykov 2006).

He focuses on an approach for creating data models and metadata models for content management systems via an interactive, iterative design process. The resulting models support a generalized metadata management process that aide in tools and software design (Zykov 2006, 1-2).

The following figure portrays Zykov's iterative development process based on a two-level conceptualization of the metadata model domains. It displays the process involved in using the various domain models to design a content management system (Zykov 2006, 2-5).

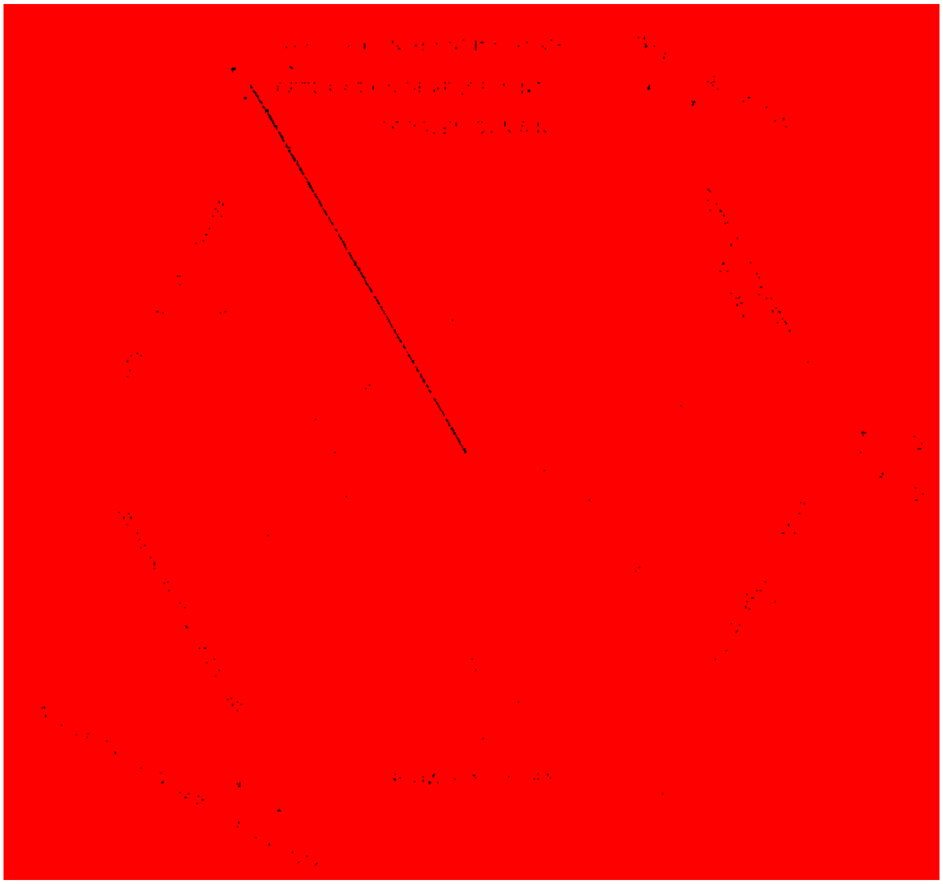


Figure 2-4 General ECM Scheme (Zykov 2006, 2)

2.5.5 Giga Information Group Enterprise Content Management Architecture

In 2002, Moore and Markham, in their discussion of a more comprehensive solution to content management, present an architecture for enterprise content management which organizations can use when evaluating potential solutions. The architecture, shown in Figure 2-5, provides a layered solution which considers the variety of content that must be managed by an organization (Moore 2002, 4).

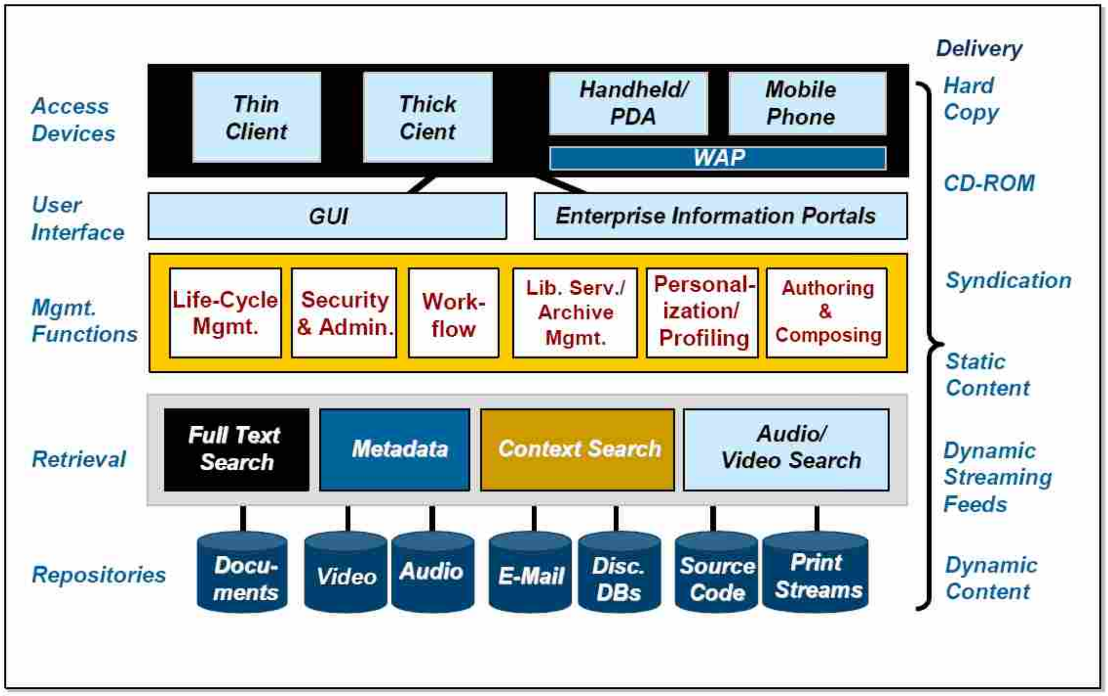


Figure 2-5 Enterprise Content Management Architecture (Moore 2002, 4)

Moore’s and Markham’s architecture presents components to provide solutions for web content management, document management, digital asset management, media set management, IDM, records management, COLD/ERM and other technologies for managing unstructured content (Moore 2002, 4).

2.5.6 The Enterprise Content Management House

Kampffmeyer, an independent consultant on enterprise content management, proposed the Enterprise Content Management House, which is shown in Figure 2-6. This model is intended to follow the process flow of content as it would move through an ECM. The house depicts the various services which would act on the content as it passes from the entry to the exit. Supporting services and clients are also represented in the figure.

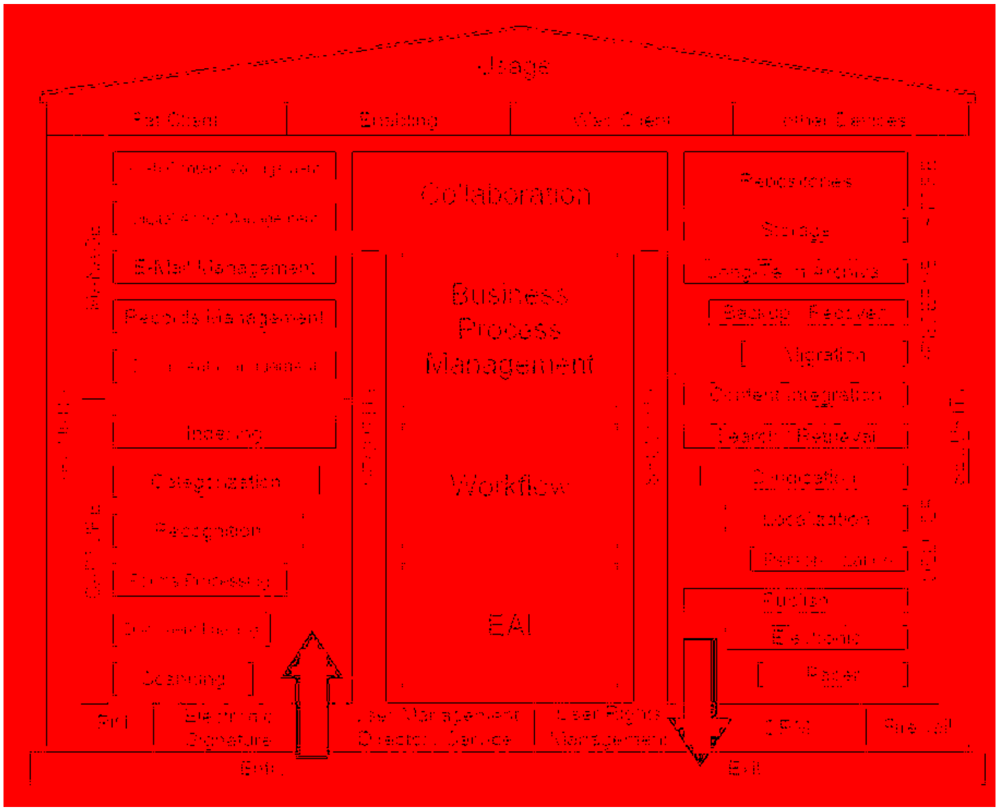


Figure 2-6 The Enterprise Content Management House (Kampffmeyer)

2.5.7 The Doculabs’ ECM Reference Architecture

Doculabs, a private ECM consulting company has allowed their ECM Reference Architecture model diagram, Figure 2-7, to be published by EMC. This figure depicts the services that a service-oriented and event-aware architecture would expose (Doculabs 2004).

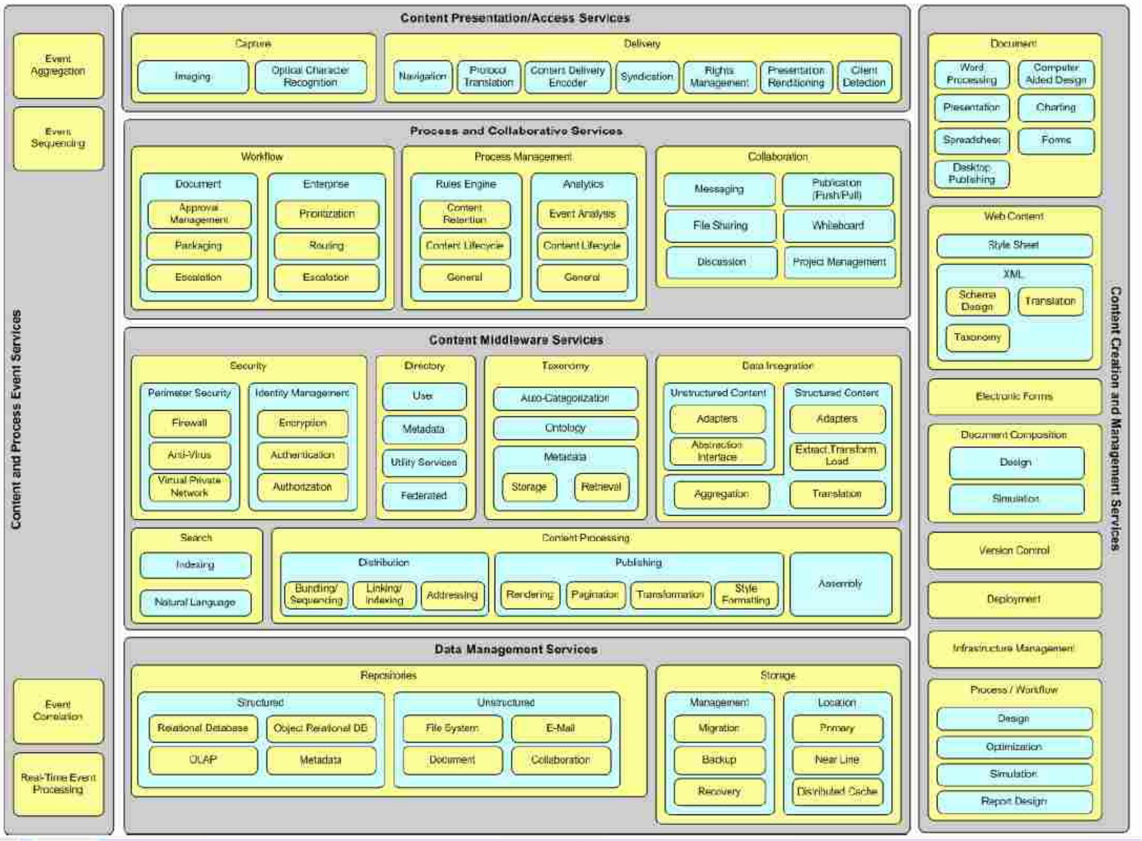


Figure 2-7 Doculabs' ECM Reference Architecture

2.5.8 The Need for an Additional Model

None of the models reviewed provided a sufficient level of detail to help identify component types and functional components for the practical development of a CMS. The models presented previously are at too abstract a level or do not address the concerns relevant to CMS development. As a result, a new conceptual framework is needed to address the appropriate level of detail for the practical development of functional components in CMS.

2.6 Case Studies of CMS Adoption and Development

The following are examples of implementations of CMSs in which conceptual frameworks were either used or developed in order to evaluate and develop CMSs. The case studies discuss the required use cases and evaluations of the implemented CMSs.

2.6.1 Franklin: Web Content Management at IBM.com

In 2002, IBM.com put into production a content management system for use as its corporate portal called Franklin. Franklin attempted to use industry standards such as XML, SXLT, CSS, etc. Weitzman, et al., discuss the creation of this system as well as the phases of development and analysis for its creation and design. Franklin manages and assembles fragments of XML that are then published as web documents (Weitzman 2002).

IBM attempted to build Franklin primarily out of open source software. Yan, in his discussion of Franklin mentions the adoption of the Apache Web Server, Perl, XML, and Struts. (Yan 2005) The adoption has been successful and IBM.com continues to adopt new open-source technologies. Evaluating open-source software can be an extensive, time-consuming process. They also consider industry standards, the application API, and documentation quality. Overall they have seen net benefits from their use of open-source software (Yan 2005, 424).

2.6.2 University of Arizona Library CMS

Han discusses the analysis process that the University of Arizona Library used to select the content management system that it would implement to meet its needs for its future content-oriented projects. Their organizational goals for a content management system were improved information accuracy through central management and the currency of information, increased flexibility through the publication of information in a variety of formats, enhanced system management through directory services integration, and reduced maintenance and cost through the development of a common infrastructure (Han 2004, 356).

The University of Arizona Library divided its CMS requirements into organizational requirements, presentation requirements, access requirements, and preservation requirements. Non-functional requirements such as cost and available technical expertise were also considered (Han 2004, 357).

The University of Arizona Library performed a detailed analysis of the following CMS products: Greenstone, Fedora and DSpace (Han 2004).

2.6.3 Statoil ECM Implementation

Munkvold, et. al, discuss ECM from the perspective of an enterprise utilizing content management technology. They report on their implementation of a major ECM project at Statoil and then identify how the encountered issues relate to established areas of information systems research such as IRM, EDM, KM. (Munkvold 2006, 71). These issues are show in Figure 2-8.

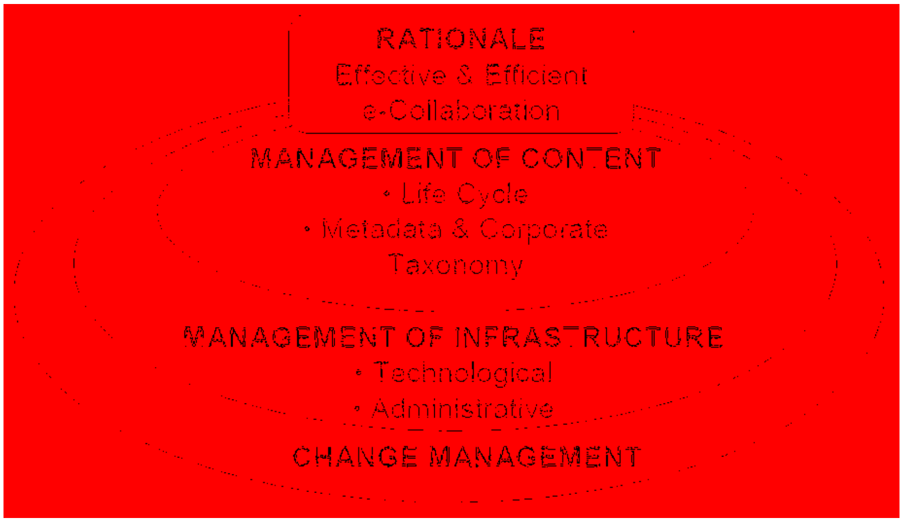


Figure 2-8 Major Categories of Contemporary ECM Issues in Statoil (Munkvold 2006, 76)

The issues encountered are organizational issues that are not necessarily related to the functionality provided by an ECM vendor's product. The issues they encountered relate to: e-collaboration, content life cycle, metadata and corporate taxonomy, technological infrastructure, administrative infrastructure, and change management. The major issues encountered are depicted in Figure 2-8 (Munkvold 2006, 76).

2.6.4 The Case Studies Did Not Have Sufficient Conceptual Frameworks

The examined use cases did not have frameworks sufficient for the practical evaluation and implementation of functional components for a CMS. The existing frameworks either did not address functionality at an adequate, practical level for functional component evalution and implementation or they did not address relevant concerns to functional component development. The lack of conceptual frameworks that provide a sufficient model for the practical evaluation and implementation of CMS functional components indicates that a new framework is needed.

3 Functional Analysis of Content Management Systems

3.1 Methodology for the Creation and Validation of the New Framework

The review of literature indicated that there is a need for a new functional framework for content management. This thesis provides a new framework which is partially validated through prototyping an architecture derived from the framework and the use cases provided in chapter 1's delimitations.

A functional analysis was performed of eight popular and prevalent CMS technologies that cover a breadth of CMS functionality. The analysis was performed through the creation of concept maps, which present a conceptual model of the available functionality. The resulting analysis was used to synthesize a list of conceptual functions, which were organized by functional group into a new conceptual framework for content management functionality. The synthesized list was used to describe the CMS functionality identified in the analysis of the existing systems in order to prove coverage. The final conceptual function list can be found in appendix A.

The contributions of this research are to create a conceptual framework, which describes the component functionality of CMSs. This framework is validated through an implementation architecture and proof-of-concept prototype, which translates the functional components from a conceptual level to a practical implementation. The prototype includes an API, which describes the interface to exercising the functionality of the components.

3.2 Introduction to the Functional Analysis of the Content Management Systems

An analysis of content management systems is begun with the identification and description of a range of content management systems. The choice of which content management systems to use in the analysis was determined by popularity, content management system type, and availability of documentation.

The analyzed content management systems are described using concept maps and a written description. Concept maps of the content management systems were made based on the available documentation. The concept maps and brief descriptions of the content management systems' functionality is provided in this chapter.

Following an individual analysis of the content management systems, a comparative analysis is performed. The results of the comparative analysis are shown in Chapter 4. The comparative analysis identifies the common functionality among the content management systems and organizes this functionality into groupings of functional components, such as document capture and document delivery.

One of the results of open-source projects is that many of the projects, especially the more popular ones, have derivative projects created as alterative solutions to the original project. The off-shoot projects are typically started by people enthusiastic about the project but want to emphasis different aspects of its development or build them around different philosophies. This thesis attempts to avoid the analysis of multiple content management systems of similar origin.

3.3 Alfresco

Alfresco is an open-source enterprise content management system available at (Alfresco Software, Inc., n.d.). Alfresco was founded in 2005 by John Newton, co-founder of Documentum, a commercial content management system, and John Powell, from Business Objects. Alfresco is an open-source alternative for enterprise content management (Alfresco Software, Inc. "About Alfresco - The Open Source Alternative for Enterprise Content Management.", n.d.).

The Alfresco documentation is available via the “Alfresco Developers” wiki at (Alfresco Software, Inc. “Alfresco Developers”, n.d.). The “Alfresco Developer” website is the source of the documentation for the analysis.

Alfresco provides functionality common to enterprise content management systems such as document management, collaboration tools, record management, web content management and image management. Alfresco is implemented in Java.

The primary functional components in Alfresco are Web Content Management, Business Process Management, the Transformation Engine, Records Management, Search, Document Management, and the Content Repository. The following figure, Figure 3-1, illustrates the basic components of Alfresco as a concept map.

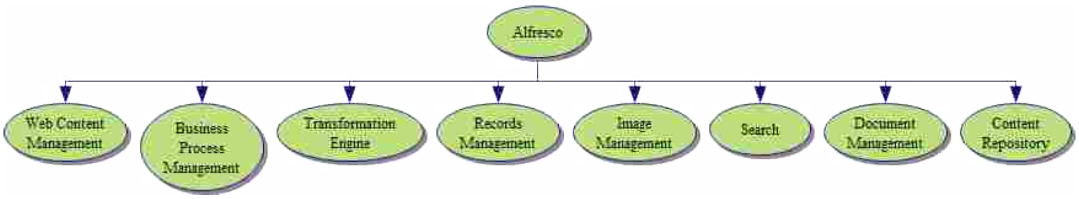


Figure 3-1 Alfresco ECM Overview

Alfresco web content management provides basic services for publishing documents online. The business process management provides a mechanism for executing a series of specific actions on a document or set of documents. The transformation engine converts a document from one type to another. Records management ensures the retention of records according to the DoD 5015.2 records management specification (U.S. Department of Defense 2007). Document management provides library services, versioning, and security for documents. Image management provides document management services specifically for images including image gallery views and image metadata extraction. The search service supports the indexing and querying of document metadata. The content repository provides the storage and access to documents.

The following is a closer examination of the eight components that comprise the Alfresco overview concept map.

3.3.1 Web Content Management

Web content management provides a means of publishing documents, including entire websites, on the World Wide Web. This includes the ability to template websites, stage documents prior to publication, manipulate content, input form data, manage dependencies, etc. This functionality is represented in Figure 3-2.

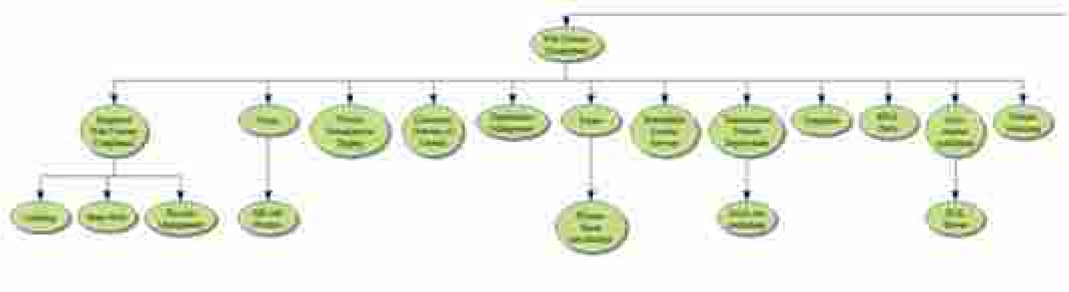


Figure 3-2 Alfresco ECM Web Content Management

In this thesis, basic support of document publication to the World Wide Web and the use of Web-based forms for metadata input and document search are explored as components of web content management in the proof of concept implementation.

3.3.2 Business Process Management

Business process management provides a means for executing a series of specific tasks on a document or set of documents. These tasks can be determined based on specific input such as the document title, the location of the document, or the document creator.

Alfresco provides a means of customizing the business process management engine by creating rules. Rule is the term Alfresco uses for the specification of a business process. Rules can be very simple common actions or be very complex. Complex rules can be programmed using the JavaScript runtime environment and API. Alfresco business process management is shown in Figure 3-3.

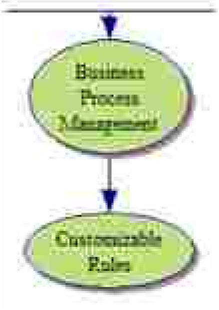


Figure 3-3 Alfresco ECM Business Process Management

3.3.3 Transformation Engine

The Alfresco transformation engine is an extendable service which takes a document of a specific type as input and outputs the content of the document as another document type. An example of this is converting a Microsoft Word document to a PDF-formatted document.

The Alfresco transformation engine allows one to create plug-ins which can extend the capabilities of the transformation engine, thus supporting the conversion between more document types. The Alfresco transformation engine components are represented in Figure 3-4.

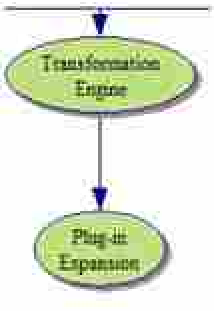


Figure 3-4 Alfresco ECM Transformation Engine

3.3.4 Records Management

Alresco records management provides a means of storing records according to the DoD 5015.2 records management specification. Records management provides a means of retaining documents of evidentiary value according to predefined rules, such as duration.

Alfresco records management includes support for document file plans, which identify a record naming convention, a retention period, metadata schema, disposition policy, etc.

Alfresco records management also supports the auto population of record metadata, the searching of record content and metadata, and the integration of record management into document workflow.

Alfresco records management can be customized using the rules-based system for business process execution.

The records management functionality available in the Alfresco ECM is shown in Figure 3-5.

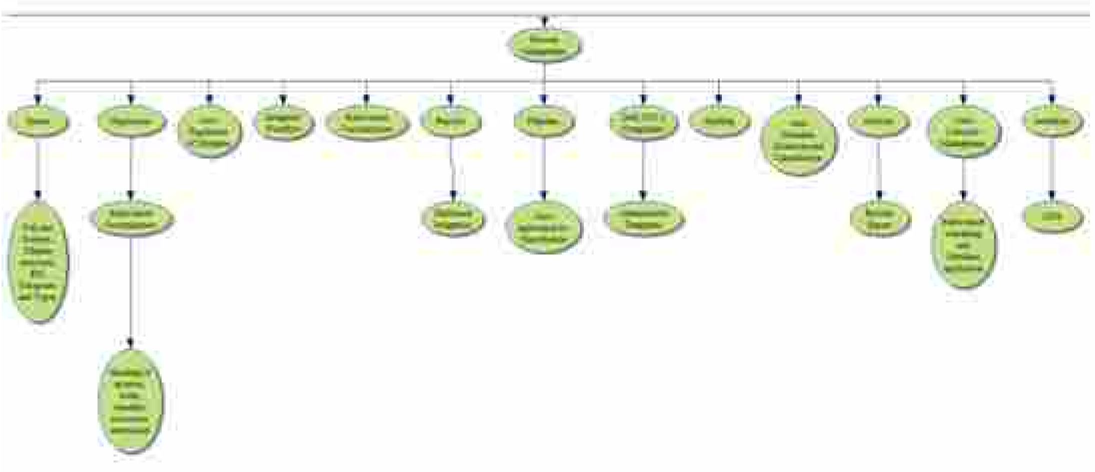


Figure 3-5 Alfresco ECM Records Management

3.3.5 Document Management

Alfresco document management provides common document management services such as content modeling and library services. Document content modeling provides a means of assigning a document a classification. This classification provides the document with a set of metadata attributes and membership in a content model hierarchy. Document library services provide a mechanism for checking out and checking in documents, tracking document versions, and managing document dependencies. Alfresco document management also supports document lifecycle management, and document-level permissions.

Alfresco document management provides several ways to interface with documents including a Common Internet File System (CIFS) share, WebDAV, and FTP.

Alfresco document management supports document collaboration. Users can be notified of document updates via e-mail and RSS feeds, and Alfresco provides forums supporting threaded discussions.

Alfresco document management provides basic, serial document routing for document review and approval. The Alfresco ECM document management functionality is shown in Figure 3-6.

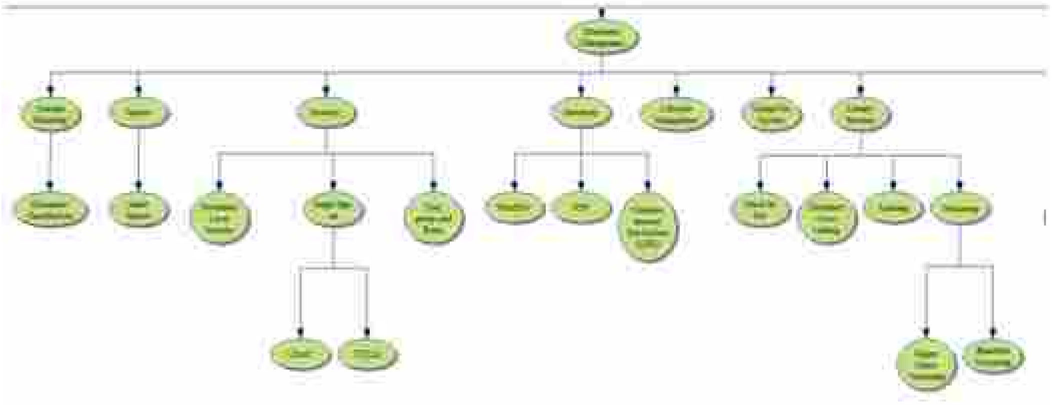


Figure 3-6 Alfresco ECM Document Management

3.3.6 Image Management

Alfresco image management provides functionality very similar to document management but is specialized for images.

Alfresco image management supports metadata extraction and auto classification of images. Image metadata and categorization can be searched, and Alfresco image management offers web services integration for the uploading and access to images. The Alfresco ECM image management functionality is shown in Figure 3-7.

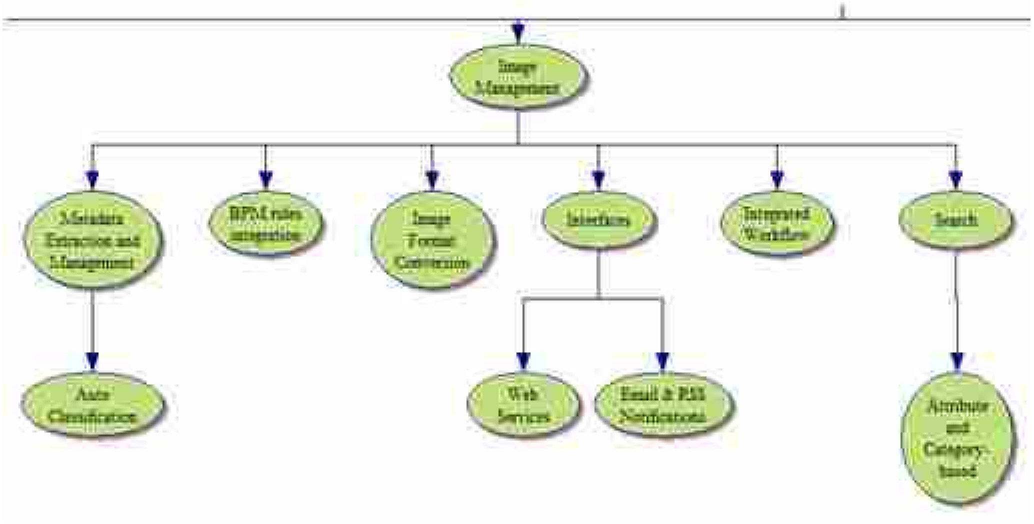


Figure 3-7 Alfresco ECM Image Management

3.3.7 Search

The Alfresco search service is integrated into the other services such as document management and records management. It supports the indexing of document metadata and content for later searching.

Alfresco supports the use of search plug-ins. Search plug-ins make it possible to provide full-text search for non-natively supported document types.

Alfresco search also supports the integration of its search results into OpenSearch-compatible clients such as Mozilla FireFox and Microsoft Internet Explorer. The Alfresco ECM Search functionality is shown in Figure 3-8.

Search
OpenSearch

Figure 3-8 Alfresco ECM Search

3.3.8 Content Repository

The Alfresco content repository is the foundation for the other services within the Alfresco ECM. The content repository ultimately provides and maintains most of the information used by the other services, such as the library services, security, document references and dependencies, document lifecycle, etc. The other services provide specialized interfaces for interacting with the content repository. The Alfresco ECM content repository functionality is shown in Figure 3-9.

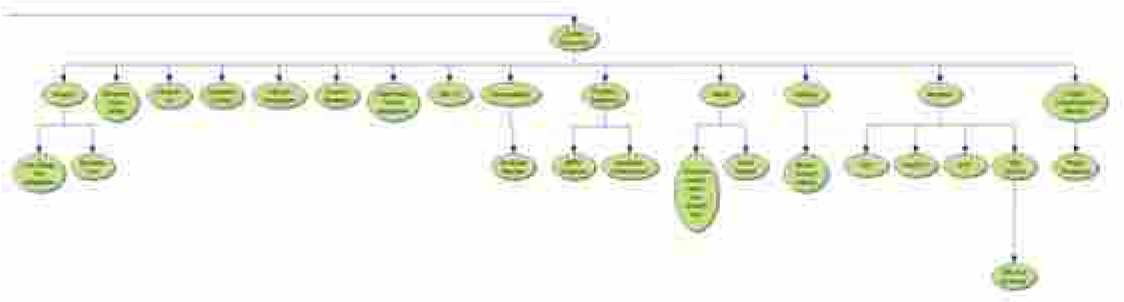


Figure 3-9 Alfresco ECM Content Repository

3.4 Apache Roller

Apache Roller was started in the Spring of 2000 by Dave Johnson (Dave Johnson, Blogging Roller, entry posted December 31, 2002). It is now an open source project supported by the Apache Software Foundation.

Apache Roller is a purpose-specific content management system designed for the management and publication of weblogs. Apache Roller supports the publication of both individual and group weblogs. It is provided as an open source project at (The Apache Software Foundation 2008). Apache Roller is implemented in Java. The documentation for Apache Roller is available at (The Apache Software Foundation 2007). This documentation is the source of information used for the following analysis. Apache Roller is used to power Sun Microsystem’s blogs.sun.com (Vyas 2008) and IBM Developerworks’ blog (Higgins 2006).

Apache Roller functionality can be divided into three basic categories. The first is administrative functionality, which allows an administrator to manage users and groups. The second category of functionality is collaborative functionality, specifically the publication of weblogs. The final category of functionality is search which is closely tied to the published weblogs.

The following figure, Figure 3-10, is the concept map for the basic functionality of Apache Roller. Given the purpose-specific nature of Apache Roller, the concept map is more detailed in its area of specialty.

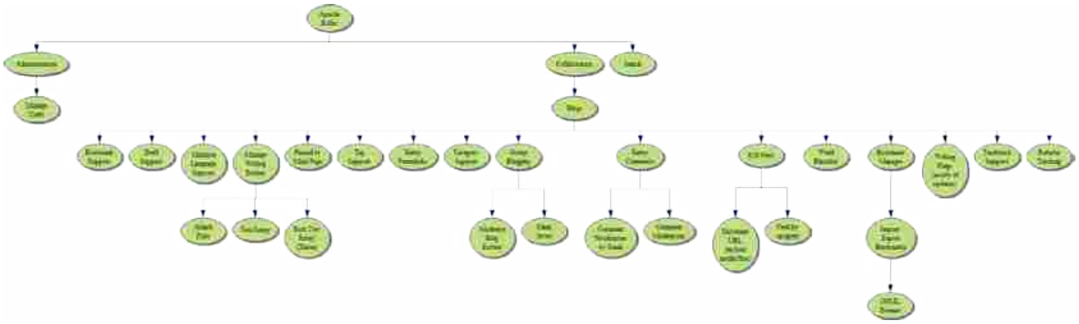


Figure 3-10 Apache Roller Functionality

3.4.1 Collaboration: Weblogs

Apache Roller is a content management system that specializes in weblog publications. A weblog is an online journal written by one or more authors. Journal entries are called weblog entries, and, at a minimum, a weblog entry consists of a title, a date, and text content. Weblog entries can also be classified with a topic, and users can request to view all entries belonging to a particular topic. A weblog entry can have a permalink, which is a static URL at which the weblog entry is located on the World Wide Web.

3.4.2 Comments

Apache Roller supports the ability to publish user-submitted comments with a given weblog entry. Moderation allows an administrator to review comments before they are published with the weblog entry.

3.4.3 Advanced Publishing

Apache Roller provides additional, sophisticated publishing tools. Supported functionality includes entry publication in multiple languages, entry drafts, file attachments, tagging, and rich text editing.

Entry drafts allow an author to save an entry without publishing it to the weblog.

Tagging allows an author to assign tags to a particular entry. A tag provides a categorization to an entry. A weblog entry may have multiple tags.

Words and phrases can be blacklisted from publication by Apache Roller. Blacklisting applies to comments, trackbacks, and referrers.

3.4.4 Syndication

Apache Roller provides syndication through a standard RSS feed mechanism. Weblog entry referrers and trackbacks are also supported. Trackbacks are an ad-hoc standard, which allows blog entries to be alerted about references to the original entry's content.

3.4.5 Group Management

Apache Roller supports the publication of weblog entries by groups of authors. In this case a particular weblog may have multiple authors who contribute entries. In a group publishing situation, entries can be moderated, meaning they must go through an approval process before they are published. Potential entry writers can be invited via an integrated email-based invitation system.

3.4.6 Search

A search system is integrated into Apache Roller to identify all weblog entries and comment text that match a particular query.

3.5 Drupal

Drupal is an open source project which serves as a foundation for content management. Started in 2000 by a University of Antwerp student named Dries Buytaert, Drupal was developed to allow him and his friends to share and discuss noteworthy things (Drupal.org, "History: About the Drupal Project." 2009).

The documentation for the Drupal content management system can be found on the Drupal website at (Drupal.org, "Drupal Handbooks" 2009). This documentation is used as the source for the following analysis of Drupal.

Drupal provides the basic services needed for a content management website. The Drupal functionality can additionally be extended via modules. The concept map in Figure 3-11 examines at the most common functionality available for Drupal, including its most popular modules. Drupal is implemented in PHP.

The concept map in Figure 3-11 breaks the Drupal functionality into two main categories: administrative and content management.

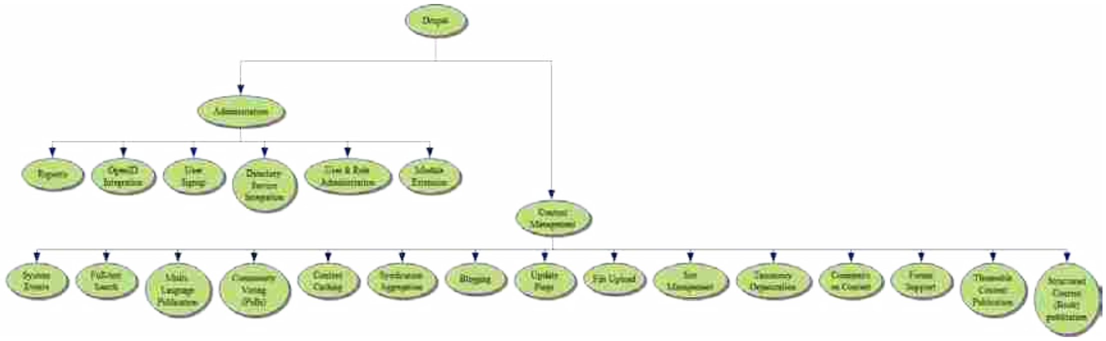


Figure 3-11 Drupal Functionality

3.5.1 Administrative Functionality

The administrative functionality of Drupal, shown in Figure 3-12, is focused on user management and module management.

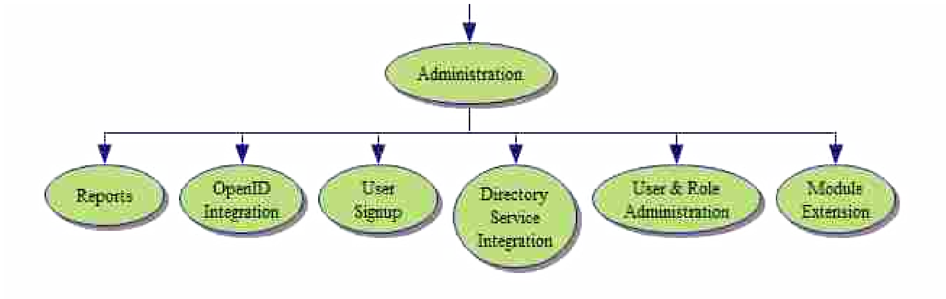


Figure 3-12 Drupal Administrative Functionality

Drupal supports the ability to either manage users within Drupal, itself, or integrate with a directory service or an OpenID provider. When managed by Drupal, users may also sign themselves up for accounts. Drupal also supports the concept of user roles.

Drupal administers can run reports on user and system activity. Administrators can manage the modules, which Drupal uses to provide its functionality.

3.5.2 Content Management

The concept map shown in Figure 3-13 identifies the content management aspects of Drupal, which include foundational functionality such as full-text search and user-end functionality such as the publication of weblogs.

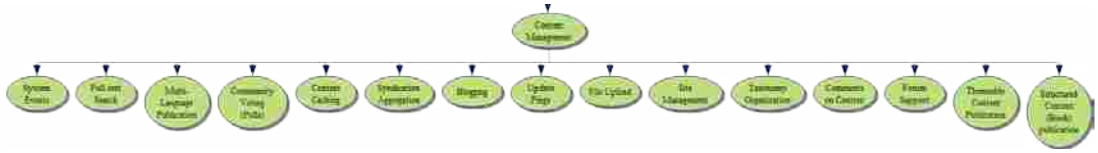


Figure 3-13 Drupal Content Management Functionality

3.5.3 Foundational Content Management Services

The foundational content management services Drupal provides include system event notifications, full-text search, content caching, file uploading, multiple-language publication, content themes, and taxonomy management. The functionality provided by these services can be used by other modules.

The Drupal system-event notification service allows modules to provide events and register for event notifications.

The full-text search functionality provides a search engine that modules can use to index and query text.

The Drupal content caching mechanism will store web-published content in a static form that can be efficiently delivered to requesting clients.

Drupal provides a mechanism that allows modules to upload and store files.

The Drupal web content publication service manages the publication of content in multiple languages. This service also allows themes to be applied uniformly across multiple published documents.

Drupal also provides a service that manages content taxonomy. This service manages tags and categories that can be applied to specific pieces of content.

3.5.4 Common Functionality

The Drupal common functionality makes use of the foundational services. Drupal supports community voting (polls), blogging, and discussion forums as commonly used modules, and Drupal will publish structured and unstructured documents. Content can have user attached comments.

Drupal will also subscribe to, aggregate, and publish common syndication formats. Drupal supports syndication including an update notification mechanism (update pings) to alert interested parties of content updates.

3.6 Moodle

Moodle is an open source content management system oriented towards educational environments. Moodle was started by Martin Dougiamas who designed Moodle while working on his Ph.D. at Curtin University of Technology in Perth, Australia. He worked as a WebCT administrator and developed Moodle to address the shortcomings of the Blackboard and WebCT systems (Tanczos 2006) (Martin 2004, under “M-2”).

Moodle is a course management system, and much of the content functionality is specific to education. Moodle is available from (Moodle, n.d.). Moodle is implemented in PHP.

The Moodle community maintains a documentation website at (MoodleDocs, n.d.). This documentation is used as the source of information for the following analysis of Moodle. The Moodle concept map is shown in Figure 3-14.

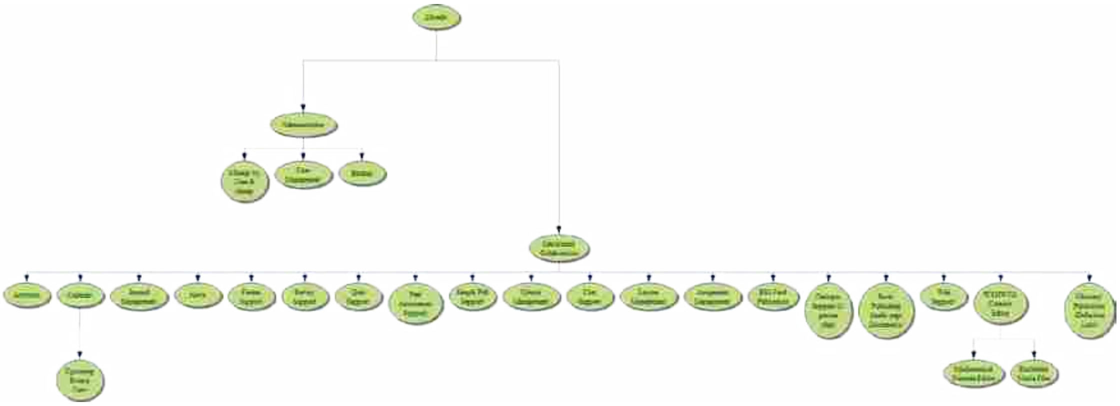


Figure 3-14 Moodle Functionality

Moodle functionality is divided into two categories: administration and educational collaboration.

3.6.1 Administration

The Moodle administrative functionality manages users and groups, including their classroom membership and classroom roles. System backups are managed through the administrative interface.

3.6.2 Educational Collaboration

Moodle provides extensive educational collaboration functionality. Moodle provides many common collaborative features such as discussion forums, live chat between users, RSS feed publication, calendaring, surveys, and wiki support.

Moodle also provides education-specific collaborative features such as on-line quizzes, peer assessments, lesson materials management, assignments management, and activities management. Student assignments and assignment completion can be tracked within Moodle.

3.7 OpenCms

OpenCms was launched in 1999 as an open-source alternative to commercial content management systems. OpenCms is supported by Framfab, a Swedish Internet consulting company (McGrath 2002). OpenCms is a content management system intended to make it easy to publish documents and entire websites on the World Wide Web. OpenCms is an open source project available at (Alkacon Software GmbH. "OpenCms: Professional Content Management.", n.d.). OpenCms is implemented in Java.

OpenCms provides a variety of downloadable documentation, which can be found at (Alkacon Software GmbH. "Download the OpenCms Documentation.", n.d.).

Figure 3-15 shows the concept map for OpenCms. The major areas of functionality are divided into the following categories: document management, content management, and administration.

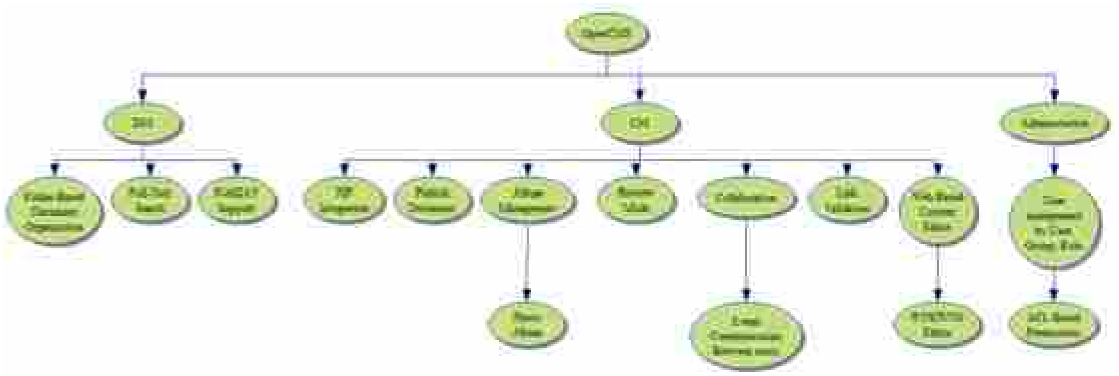


Figure 3-15 OpenCms Functionality

3.7.1 Administration

The administration functionality of OpenCms supports management of users, groups, and roles. User management includes ACL-based permission management.

3.7.2 Document Management

The document management functionality of OpenCms provides full-text search of documents and folder-based document organization. OpenCms document management also supports the access and modification of documents via the WebDAV interface.

3.7.3 Content Management

OpenCms content management supports the publication of documents on the World Wide Web. Document templates can be created via JavaServer-Pages-based templates. Web publication features include the ability to edit and preview content prior to publication. OpenCms will validate links between documents and provides an album view for images, and it provides a visual WYSIWYG content editor for the creation of content for web publication.

OpenCms supports collaboration in the management of e-mail groups and the delivery of email communication between users.

OpenCms also supports the management of digital image content. OpenCms stores images and organizes them based on albums.

3.8 Plone CMS

Plone CMS is an open source content management system for publishing documents on the World Wide Web. Plone is available at (The Plone Foundation and others. "Plone CMS: Open Source Content Management." 2009) and is implemented in Python. The Plone CMS documentation used for this analysis is available at (The Plone Foundation and others. "Plone Manuals" 2009).

Figure 3-16 shows the Plone CMS concept map identifying the major functionality. The major functionality is divided into web content management and administrative functionality.

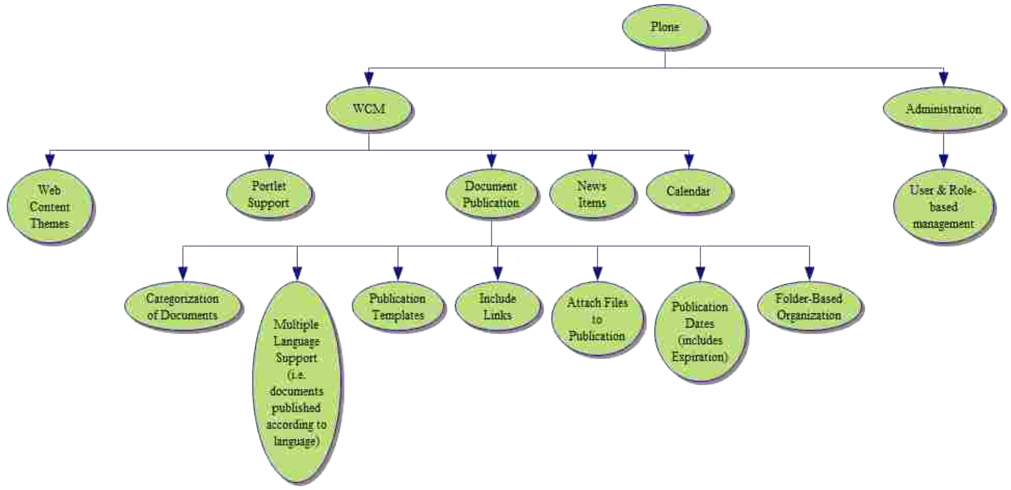

Plone
WCM
Administration
Web Content Themes
Portlet Support
Document Publication
News Items
Calendar
User & Role-based management
Categorization of Documents
Multiple Language Support (i.e. documents published according to language)
Publication Templates
Include Links
Attach Files to Publication
Publication Dates (includes Expiration)
Folder-Based Organization

Figure 3-16 Plone Functionality

3.8.1 Administration

The Plone CMS provides basic administrative functionality for the management of users and their associated roles.

3.8.2 Web Content Management

The web content management functionality of the Plone CMS supports the publication of documents to the World Wide Web. Plone supports the management of documents in multiple languages. Templates can be used for customizing the presentation of documents.

The Plone CMS can automatically publish content by date-based time frames and includes the ability to automatically expire content.

Plone also supports website-wide themes. Themes allow Plone to provide a consistent look and feel across all the content of a specific website.

Plone can publish news items and calendar items.

Documents are organized within the Plone CMS according to folder and content categorization.

3.9 SpringCM

SpringCM is a content management system provided as a service for a regular subscription fee. It can be found at (SpringCM, n.d.).

The overview of the functionality provided by SpringCM is shown in Figure 3-17. This functionality is divided into the following categories: capture, workflow and collaboration, document delivery, web services integration, security, administration, and document management.

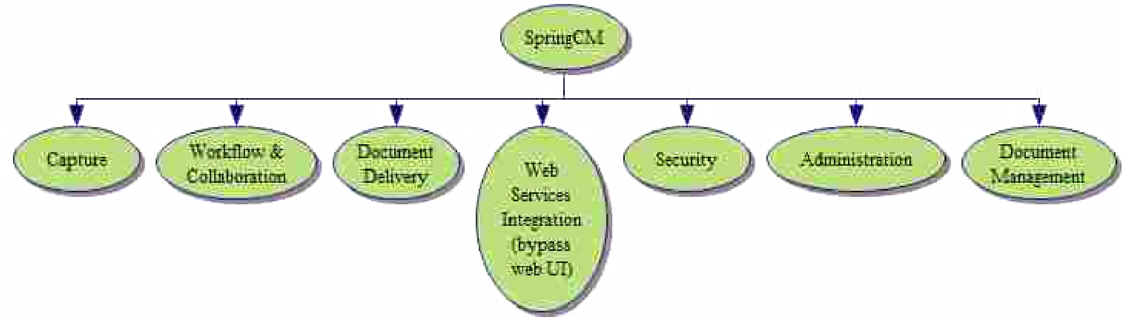


Figure 3-17 SpringCM Functionality

3.9.1 Document Capture

The SpringCM provides ways of capturing content, which are shown in Figure 3-18. Capturing data can be accomplished using interface APIs such as FTP, browser upload, and email. SpringCM also provides several client-side programs for capture to

SpringCM such as a print driver, scanner driver, fax modem, and a Microsoft Office plug-in. As part of the capture process, documents can be converted to text using OCR, converted to PDF, have metadata automatically extracted and manually assigned, and be automatically assigned to the appropriate individual.

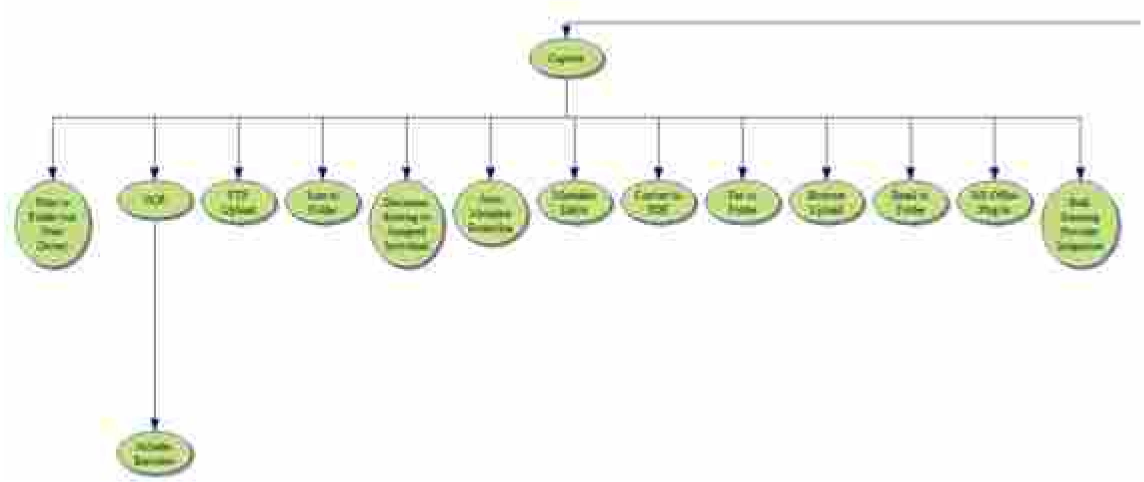


Figure 3-18 SpringCM Capture Functionality

Once a document has been captured, the metadata can be automatically extracted. Based on the available metadata, the document can be automatically routed to the correct individual or group. The metadata can be edited manually by the user.

3.9.2 Workflow and Collaboration

SpringCM provides support for workflow-based collaboration on documents. The SpringCM supported functionality, shown in Figure 3-19, includes document routing and approval processes that include both serial and parallel routing. Document routing assignments can have specific time frames within which they must be completed.

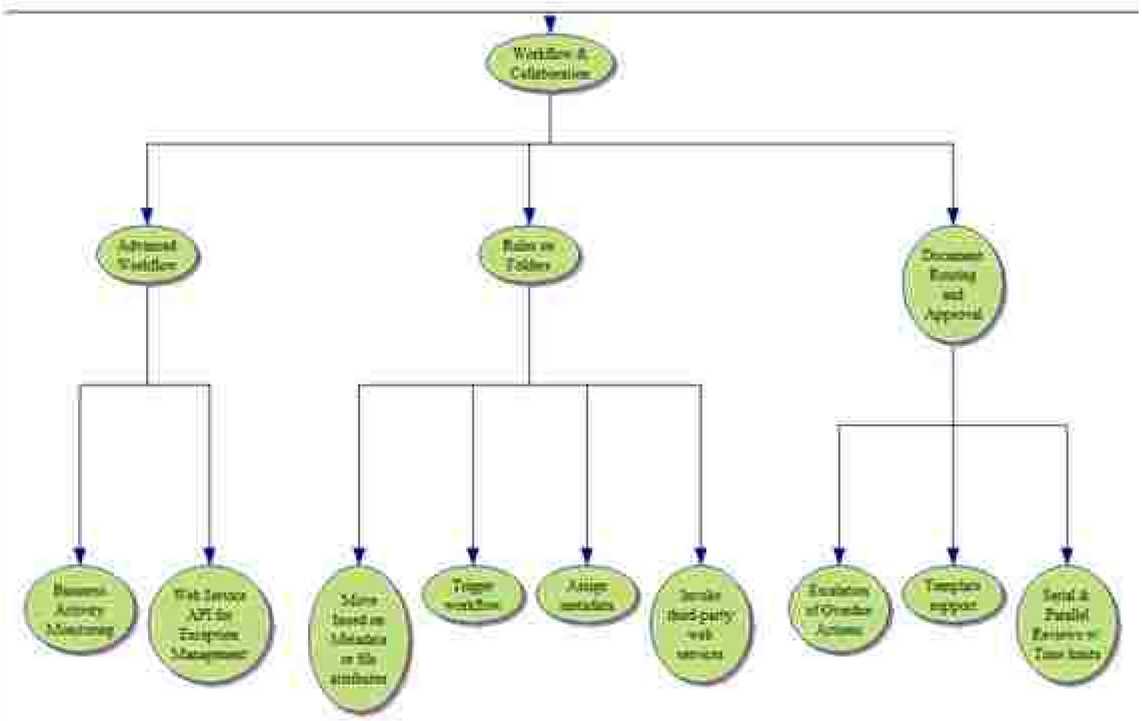

Workflow & Collaboration
Advanced Workflow
Rules on Folders
Document Routing and Approval
Business Activity Monitoring
Web Service API for Exception Management
Move based on Metadata or file attributes
Trigger workflow
Assign metadata
Invoke third-party web services
Escalation of Overdue Actions
Template support
Serial & Parallel Reviews w/ Time limits

Figure 3-19 SpringCM Workflow and Collaboration

SpringCM supports document processing rules on folders allowing metadata assignment or workflow initiation when certain events occur. Third-party web services can also be invoked according to rules on folders.

SpringCM also supports workflow features such as a web service API for exception management and business activity monitoring of ongoing processes.

3.9.3 Document Delivery

SpringCM provides support for several forms of document delivery as shown in Figure 3-20. This includes publication of documents to the World Wide Web as the original document or PDF and the delivery of documents via e-mail, fax, and mail. Documents can be guaranteed authentic via electronic signatures.

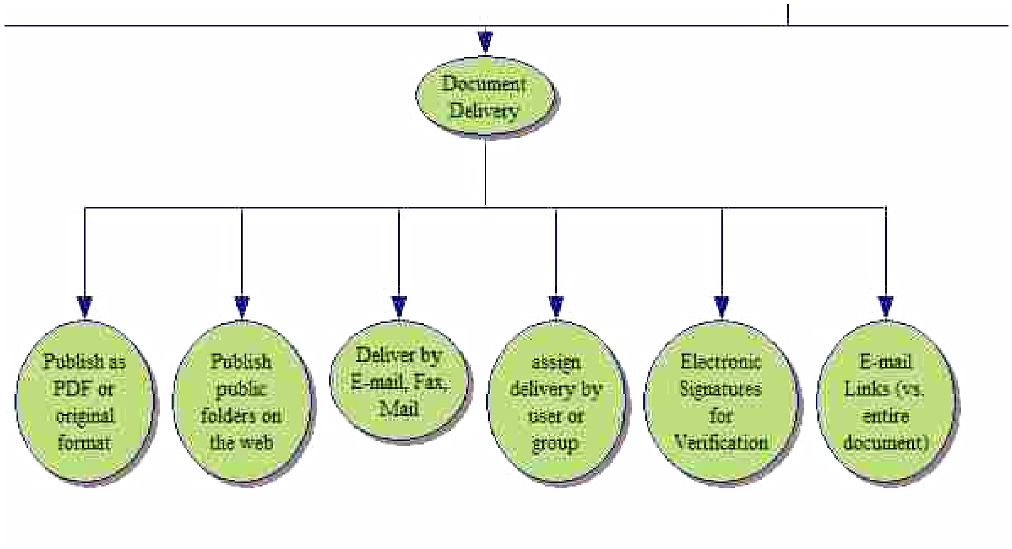

Document Delivery
Publish as PDF or original format
Publish public folders on the web
Deliver by E-mail, Fax, Mail
assign delivery by user or group
Electronic Signatures for Verification
E-mail Links (vs. entire document)

Figure 3-20 SpringCM Document Delivery Functionality

3.9.4 Web Services Integration

SpringCM provides integration with its functionality via web services. These web services allow users of SpringCM to bypass the typical web-based user interface.

3.9.5 Security

SpringCM provides document-based security and support for a variety of standards as shown in Figure 3-21. Supported security standards include NIST 800-53 security controls (U.S. Department of Commerce. "NIST Special Publication 800-53 Revision 2: Recommended Security Controls for Federal Information Systems.", 2007), Control Objectives for Information and Related Technology standards security design (Information Systems Audit and Control Association. 2009), and IT Infrastructure Library standards-driven security design (The APM Group Limited 2009).

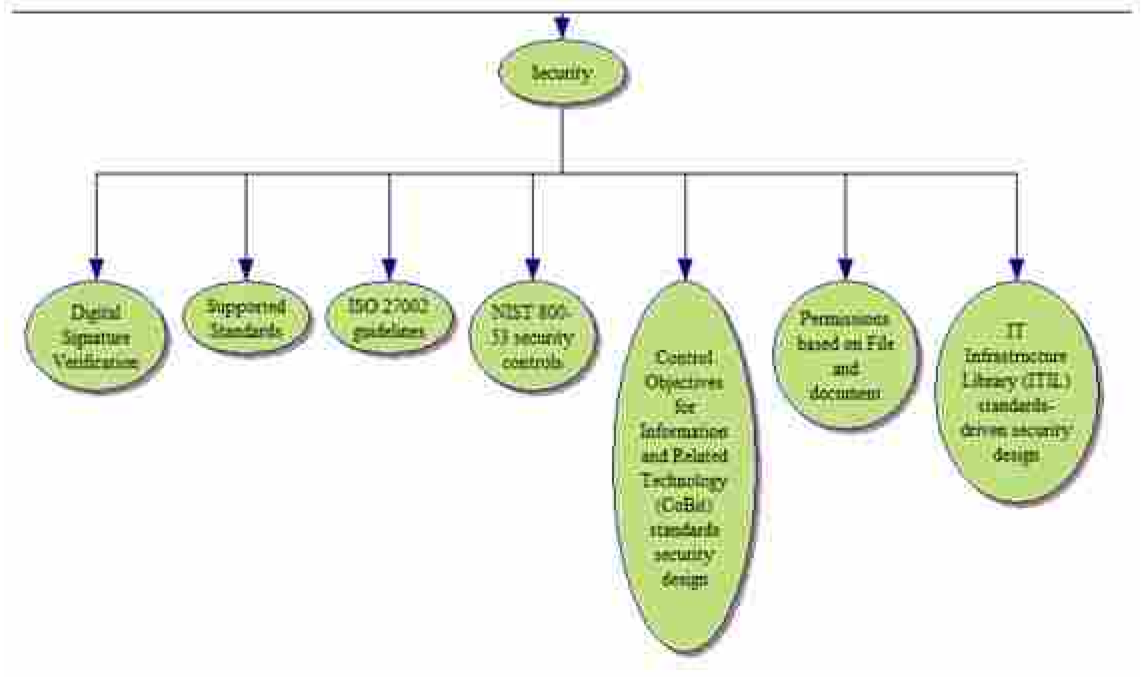

Security
Digital Signature Verification
Supported Standards
ISO 27002 guidelines
NIST 800-53 security controls
Control Objectives for Information and Related Technology (CoBit) standards security design
Permissions based on File and document
IT Infrastructure Library (ITIL) standards-driven security design

Figure 3-21 SpringCM Security Functionality

3.9.6 Administration

The administrative abilities of SpringCM, shown in Figure 3-22, allow an administrator to manage users, groups, and assigned roles. The administrator can also customize the aesthetics of standard Web user interface.

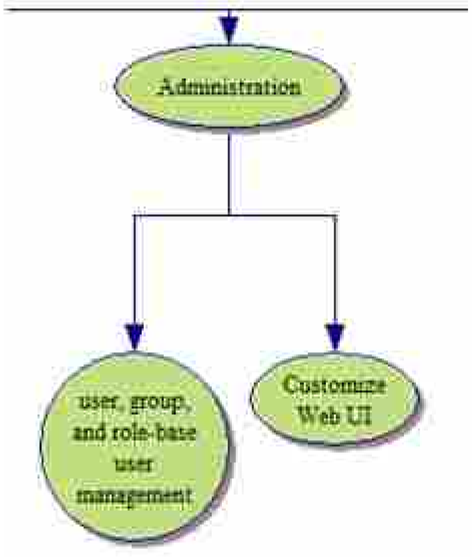


Figure 3-22 SpringCM Administrative Functionality

3.9.7 Document Management

SpringCM provides document management services, shown in Figure 3-23, such as document versioning and library services. Documents can also be annotated and be set to expire on a specific date.

SpringCM will generate reports based on document activity.

Documents can be organized within SpringCM based on folder structure, metadata, and keyword tags.

Documents can be searched for based on their metadata and their content.

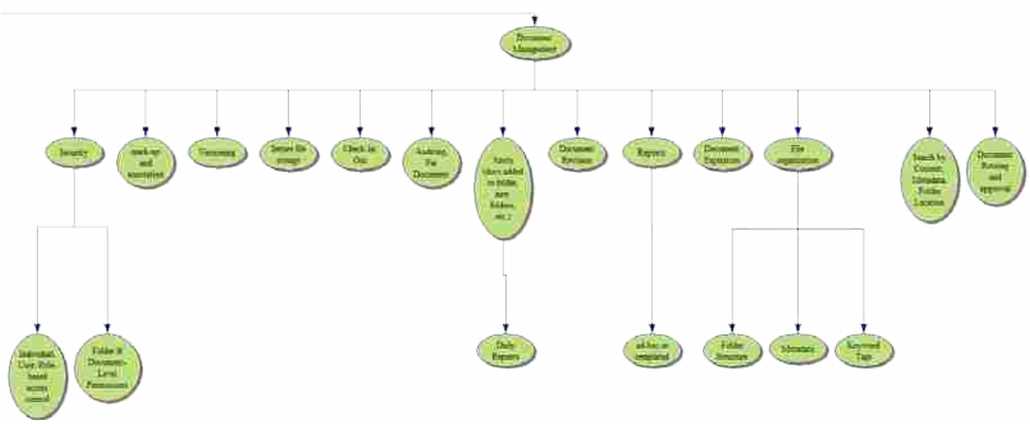


Figure 3-23 SpringCM Document Management Functionality

3.10 Symantec Enterprise Vault

The original Enterprise Vault product was designed by Digital Equipment Corporation in 1997. Enterprise Vault was made commercially available in 1999 as a universal repository and management solution for unstructured data (Brown 2007).

Symantec Enterprise Vault provides a means of capturing and archiving documents. It is provided as a commercial product from Symantec. More information can be obtained on Enterprise Vault at (Symantec Corporation 2009). The functionality provided by Enterprise Vault is shown in Figure 3-24.

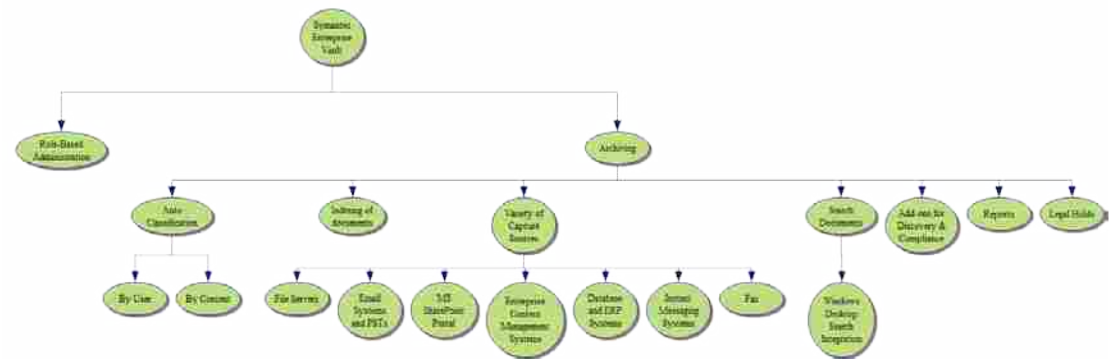

Symantec Enterprise Vault
Role-Based Administration
Archiving
Auto Classification
Indexing of documents
Variety of Capture Sources
Search Documents
Add ons for Discovery & Compliance
Reports
Legal Holds
By User
By Content
File Servers
Email Systems and PSTs
MS SharePoint Portal
Enterprise Content Management Systems
Database and ERP Systems
Instant Messaging Systems
Fax
Windows Desktop Search Integration

Figure 3-24 Symantec Enterprise Vault Functionality

3.10.1 Administration

Enterprise Vault supports role-based administration of users and their permissions.

3.10.1.1Archiving

Enterprise Vault supports the capture and storage of documents from a variety of sources including file servers, email systems, PST archives, Microsoft SharePoint, many databases, instant messaging systems, and faxes. Enterprise Vault can automatically classify documents and support the search of documents content and metadata.

Enterprise Vault provides reports on archive contents and supports legal holds and add-ons for discovery and compliance.

3.11 Analysis Results

The analysis presented in this chapter of the functional components resulted in one or more concept maps for each of the eight analyzed CMS. These concept maps provide a conceptual view of the available functionality of each CMS.

The resulting concepts maps will be synthesized in the next chapter into a comprehensive list of CMS functionality and then used to create the new conceptual framework.

4 A New Conceptual Framework of Content Management Functionality

This chapter presents the new conceptual framework for content management functionality. The framework is derived from an analysis of eight content management systems. It describes forty-five conceptual functions organized into five functional groups. The functionality derived from the analysis of the content management systems is described using the vocabulary provided by the functional framework. Coverage of the concepts in the existing systems is verified. The next chapter, Chapter 5, discusses the prototype used to validate the utility of the functional framework. The architecture demonstrates how to translate the conceptual functions from a conceptual level as described by the framework to a practical implementation.

The proof-of-concept prototype discussion in Chapter 5 includes an overview of the API used to interface with the implemented functional modules. The description of the API demonstrates how the functional modules interrelate.

This chapter presents an analysis and synthesis of the functionality of the content management systems described in chapter 3. The conceptual functionality identified in the previous chapter is presented in Appendix A, the content management system functionality chart. This information is presented in such a way that one can determine which functionality is common among which content management systems.

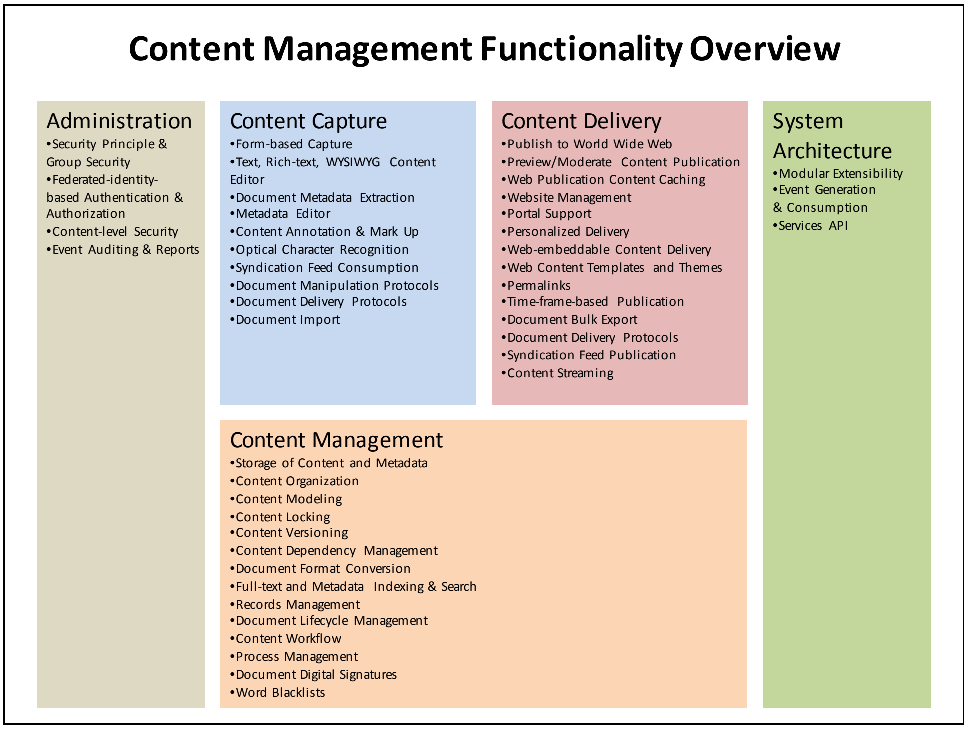
The chart presents an overview of the available functionality of the analyzed content management systems as a list of conceptual functions. This refined list represents the functional components after the redundancies in functionality between content management systems have been removed and the vocabulary rationalized.

Each functional component was analyzed to determine whether it can be decomposed to other, more basic, conceptual functions. The resulting list of basic conceptual functions is included in as Appendix A.

4.1 Functional Analysis Table

Appendix A shows the results of the content management system analysis. It enumerates the functional components and identifies in which content management systems the functionality can be found. The component type, such as whether it is identified as document management or record management, of each conceptual function is also identified, as well as a description of each. Basic and Composite Functional Components

The basic conceptual functions can be grouped into the following types: system architecture, administration, content capture, content delivery, and content management as shown in Figure 4-1. The conceptual functions identified as basic are described below and are organized by type.



Content Management Functionality Overview

Administration
- Security Principle & Group Security
- Federated-identity-based Authentication & Authorization
- Content-level Security
- Event Auditing & Reports

Content Capture
- Form-based Capture
- Text, Rich-text, WYSIWYG Content Editor
- Document Metadata Extraction
- Metadata Editor
- Content Annotation & Mark Up
- Optical Character Recognition
- Syndication Feed Consumption
- Document Manipulation Protocols
- Document Delivery Protocols
- Document Import

Content Delivery
- Publish to World Wide Web
- Preview/Moderate Content Publication
- Web Publication Content Caching
- Website Management
- Portal Support
- Personalized Delivery
- Web-embeddable Content Delivery
- Web Content Templates and Themes
- Permalinks
- Time-frame-based Publication
- Document Bulk Export
- Document Delivery Protocols
- Syndication Feed Publication
- Content Streaming

System Architecture
- Modular Extensibility
- Event Generation & Consumption
- Services API

Content Management
- Storage of Content and Metadata
- Content Organization
- Content Modeling
- Content Locking
- Content Versioning
- Content Dependency Management
- Document Format Conversion
- Full-text and Metadata Indexing & Search
- Records Management
- Document Lifecycle Management
- Content Workflow
- Process Management
- Document Digital Signatures
- Word Blacklists

Figure 4-1 Content Management Functionality Overview

4.2 System Architecture

Content management systems provide extensible functionality through the ability to augment the system with modules. Integration capabilities are made possible through event generation and event consumption APIs.

4.2.1 Modular Extensibility

A system architecture, which supports the extension of services via modules or plug-ins, enhances customizability and integration of content management systems with business processes. Extensible aspects include plug-ins for file conversion and modules that augment system functionality. Modules can also provide functionality for legal compliance and legal holds.

4.2.2 Event Generation and Consumption

Many content management systems provide an event management mechanism, which allows components within the content management system to interact and communicate with each other. System events provide a loosely coupled mechanism for coordination.

4.2.3 Services API

Integration with the content management system is simplified when a standards based API is provided. Use of an API maintains the reliability of the system data and prevents the bypass of mechanisms such as security and business process actions.

4.3 Administration

Administrative functionality is typically required for the operation of a content management system. This functionality includes security principle and group security, authentication services integration, fine-grained security, and services APIs.

4.3.1 Security Principle and Group Security

The most basic level of content security is based around security principle and group-based security. Content management systems typically include the concept of a content user (a security principle) who views and acts on content. A group is a collection of security principles. Membership in a group gives the security principle the privileges of the group in addition to those which the security principle already has assigned to them.

4.3.2 Federated-identity-based Authentication and Authorization

Federated-identity-based systems such as directory services can simplify the management of users and integration of content systems. Federated identity systems are responsible for providing their own security principle management.

4.3.3 Content-level Security

Permission security on individual content-containing documents provides greater flexibility to meeting user needs than when permission management is limited to more inclusive organizational groupings, such as a folder.

An ACL-based permission scheme provides the greatest potential flexibility in specifying content permissions.

4.3.4 Event Auditing and Reports

The content change and user and system actions which occur within the system should be tracked. Ideally, audit information should be associated with version information; so that what user or process has affected a piece of content in a particular way can be identified. Additionally, non-destructive actions should be audited, such as which user viewed a particular piece of content.

Audit reports allow for the aggregation of data about particular events. This supports purposes such as security investigations and trend analysis.

4.4 Content Capture

Capture functionality is the process of getting data into the content management system. This includes the capture of the document content and the associated metadata.

4.4.1 Form-based Capture

Form-based data capture supports the input and customization of structured or semi-structured data. Data entry forms can be customized using a form description standard such as the W3C XForms standard (Pemberton 2009).

4.4.2 Text, Rich-text, WYSIWYG Content Editor

Plain and formatted text and other types of media can be captured through editors that support features such as the formatting of text and the inclusion of digital media, such as images.

4.4.3 Document Metadata Extraction

Metadata extraction for documents provides a means of capturing a document's metadata attributes. This metadata includes the document type, relevant dates, and metadata attributes that may be embedded within the document.

4.4.4 Metadata Editor

Metadata editors support the modification of the metadata attributes associated with a particular document.

4.4.5 Content Annotation and Mark Up

Document annotation and mark up is an additional form of content metadata editing. This includes the ability to associate metadata with a particular portion of a piece of content.

4.4.6 Optical Character Recognition

Optical character recognition (OCR) provides an automated means of capturing the textual content of image-formatted content. Scanned documents typically lose their textual content if the resulting document is an image. OCR technology can be used to identify the text that exists in the image and provide it in a more convenient, machine-readable manner.

An alternative way of categorizing OCR is as a part of format conversion. According to this categorization, OCR is being used to convert an image-based document to a document type which contains textual content.

4.4.7 Syndication Feed Consumption

Syndication feeds provide a mechanism for sharing information with interested subscribers. This information can include system events and content. An example syndication feed mechanism is RSS.

4.4.8 Document Manipulation Protocols

Many protocols exist which allow the management of documents on the content management system. WebDAV, FTP, and CIFS are examples.

4.4.9 Document Delivery Protocols

A variety of protocols and methods exist for capturing data to a content management system. These protocols include HTTP for file upload, fax, email and proprietary software that support print and scan-based interfaces.

4.4.10 Document Import

Content management systems support the importing of data in a variety of formats. Sources include Microsoft Outlook PST and OST, DBMS, and Microsoft

SharePoint formats. The ability to import a document allows a content management system to extract content from a given document.

4.5 Content Delivery

4.5.1 Publish Content to the World Wide Web

Content management systems can use the Internet and the World Wide Web to make a variety of content formats available. Web publishing supports the publication of structured content and the representation of content as multi-page documents.

4.5.2 Preview and Moderate Content Publication

Content can be reviewed by a moderator prior to publication. This may happen as part of a pre-defined workflow.

4.5.3 Web Publication Content Caching

Content is often aggregated via templates on an as needed basis. The results of this process can be stored for more efficient retrieval at a later point in time.

4.5.4 Website Management

Entire websites are managed by content management systems. Some content management systems will manage entire versions of websites with features such as website snap shots. Website management features also include support for the transactional deployment of content.

4.5.5 Portal Support

Portal publication systems support a means of displaying a variety of small segments of web pages. Content management systems provide both portal publication and a means of displaying portal page fragments in other portal publication systems.

4.5.6 Personalized Delivery

Content publication can be personalized based on user preferences and other associated metadata, such as location or user credentials.

4.5.7 Web-embeddable Content Delivery

Embeddable content is content which can be inserted or included in other web sites or publication formats not provided by the content management system providing the embedded content.

4.5.8 Web Content Templates and Themes

Web content is provided a consistent look and feel through the use of templates and themes. Templates provide a uniform layout to content. Themes provided a consistent look and feel to multiple pieces of content.

4.5.9 Permalinks

Permalinks are static URLs that content management systems provide to identify a piece of content. Permalinks are an easily shareable reference to content.

4.5.10 Time-frame-based Publication

Time-frame-based publication enables content to be published only during the specified time. Outside of the given time frame the content is unavailable in the specified publication medium.

4.5.11 Document Bulk Export

A bulk export feature of content allows for the archival and backup of content.

4.5.12 Document Delivery Protocols

Email can be used as a means of delivering content, notification of events, and as a means of referencing content not included in the email. An email content reference might, for example include a permalink, which the user can click on to retrieve a piece of content.

4.5.13 Syndication Feed Publication

Syndication feeds provide a means of sharing data between systems. A content management system can provide a syndication feed to make other systems aware of events and to access content. An example syndication format is RSS.

4.5.14 Content Streaming

Content streaming allows portions of content to be requested. This allows large files, such as video media, to be requested as multiple smaller segments.

4.6 Document Management

4.6.1 Storage of Content and Metadata

Content and associated metadata is stored by a content repository within the content management system. An item of content can potentially have an arbitrary number of pieces of metadata associated with it.

4.6.2 Content Organization

Content can be organized via a variety of means within the content management system to provide the greatest flexibility possible. In some cases multiple organizational schemes may be applied to the same set of content. The most popular organizational methods include placing content in a folder hierarchy and creating groups of content called spaces. Other metaphors may be used for organizing content such as photo albums.

4.6.3 Content Modeling

A classification scheme is often used to ensure the content has an appropriate class or type assigned to it and that it has the necessary metadata attributes. More sophisticated content modeling uses classification schemes that are hierarchical in nature, which allows a child class to also consist of the metadata attributes of its parent class. Content modeling is used to ensure the data integrity of content.

4.6.4 Content Locking

Content locking allows an entity within the content management system to temporarily reduce the permissions of a given file. This ensures concurrency while a piece of content is being acted upon. For example, a user may prevent a piece of content from being modified by other users while the user, who initiated the lock, is editing the content. Another example is the reporting system may lock a large collection of content from being modified while a report is being generated on the content metadata.

4.6.5 Content Versioning

Content versioning allows a content management system to keep track of multiple versions of a single piece of content. This functionality potentially serves a variety of purposes, such as keeping track of a document history for auditing purposes or providing a means of managing a document which is being published to multiple languages.

4.6.6 Content Dependency Management

Content dependency management provides a means of ensuring references between content are valid. References between content is also referred to as cross linking.

The most common case of a reference dependency is one piece of content requiring that another piece of content exist. An example of this would be a document rendition, which should not exist without the original document. The original document could not be deleted until all of its renditions have also been deleted.

Dependency management may be used to ensure that all of the hyperlinks in a Web-published document reference other relevant content prior to the document being eligible for publication.

4.6.7 Document Format Conversion

Document format conversion provides a means of converting a document from one document type to another. For example, a user may want to publish a MS Word document to the web but choose to do so as a non-editable PDF. The desired document rendition is generated by the content management system.

For the greatest possible flexibility, document format conversions systems are extensible via plug-ins. Plug-ins allow for the addition of supported document types.

4.6.8 Full-text and Metadata Indexing and Search

Full-text and metadata search allows content to be identified based on the textual content of the content and the content metadata. A search query is performed against a set of content; the content which meets the query criteria is reported as the result.

4.6.9 Records Management

Records management ensures that a content management system retains content according to a specific retention schedule. Records management activities are dictated by a file plan, which identifies things such as content metadata requirements, a retention schedule, a disposition strategy, a lifecycle, and a review schedule.

The government specification for records management, DoD 5015.2, can be found at (U.S. Department of Defense 2007)

4.6.10 Document Lifecycle Management

Each document has a single lifecycle associated with it. The document lifecycle is used to identify the current state of the document within the content management system. The document lifecycle is dependent on the management process for a particular document. For example, a document pending review prior to publication may be in a review for publication state in its lifecycle. Once the document is approved, the document will be published to the World Wide Web and be identified as being in a published state in its lifecycle. Once the document is no longer published it may be put into an archived state in its lifecycle while it awaits disposal.

4.6.11 Content Workflow

Content workflow is the management of the movement of content between users through a pre-defined process. Users in the process may edit the content or provide feedback, review, or approval

4.6.12 Process Management

Process management handles the automated processing of content. Process management is often customizable via a rules-based system or scripting environment.

Process management may perform actions such as auto-classification of documents, format conversion of captured documents, or initiation of a document workflow.

4.6.13 Digitally Sign Documents

In order to verify the authenticity of a document, content management systems may digitally sign a document. A digital signature may be used to verify the document has been received in its original state and that the document was received from the intended party.

4.6.14 Word Blacklist

A word blacklist provides an automated way of censoring content. This service can be used to process content for appropriateness prior to publication.

4.7 Composite Functional Components

The following forms of content management functionality are specialized uses of the previously described content management functionality referred to as composite functional components. A composite functional component is a specialized use of one or more basic functional components.

4.7.1 Calendar Management

Calendar management supports the management and publication of calendar entries. Calendar entries are a type of specialized content, which can be published to users based on their personalized preferences. Means of publishing the calendar entries include web publication and publication via syndication feed formats. The content management system security ensures unintentional publication of calendar entries.

4.7.2 Journal Management

Content does not necessarily need to be publically or widely published. A content management system can exist simply to store and manage the content in case it is needed. An example of this content would be journal management in a classroom environment. The student journal entries can be captured using functionality such as rich-text capture. The journal content is secured using the security functionality and published only to those authorized to access the content on an as-needed basis.

4.7.3 News Publication

Publication of news items is a form of content publication. News publication provides news snippets, which are pieces of content that are published based on their associated metadata, which may include time-frame, topic, or other relevant factors. Means of news content publication includes Web publication and syndication feeds.

4.7.4 Quiz, Peer Assessment, and Survey Support

Quiz, peer assessment, and survey functionality are all examples of content that is published to specific users after which a response to the published content is captured. Once a response is captured, the content management lifecycle can published the captured responses to the appropriate individuals. The captured responses may be structured data, such as true-false and multiple-choice answers, and unstructured data, such as essay responses.

4.7.5 Chat Support

Chat is a form of collaborative communication in which two or more individuals are communicating in real time by asynchronously publishing pieces of content that are viewable by those involved in the conversation. Chat support is content publication in which the individual chat messages are captured, there is minimal content processing, and the content workflow directs the publication of the content to those involved in a particular conversation. Specialized clients are required to support the asynchronous capture and publication of the chat content.

4.7.6 Course, Lesson, and Assignment Management

Course, lesson, and assignment management is the publication of specialized content that is typically intended for a classroom environment. Course management features of a content management system would track and publish content and metadata related to courses, such as availability, instructor information, and syllabus. The lesson management aspect of content management captures, manages, and publishes content relating to specific lessons for a course. An individual lesson may include a collection of relevant content, such as lesson materials or a lesson discussion forum. Finally, assignment management can publish the assignments for a given lesson or an entire course and provide a means of capturing and then managing the assignment response.

4.7.7 Wiki Support

A wiki provides capture and publication of document content which one or more people may view and edit. A wiki can track multiple versions of a given piece of content and identify who made what changes to the content, as well as when the changes were made. Wikis make it convenient for many people to contribute content using a very informal workflow and to roll back to previous versions of content as needed.

A wiki can be seen as composite content management functionality in that it is capturing storing and publishing multiple versions of content. A wiki provides purpose-specific interfaces for capturing the content and for viewing the differences between versions. The content management system security functionality enables the capture of metadata related to who edited the content and prevents unauthorized authoring of content.

4.7.8 Glossary Publication

A glossary is consists of multiple words and each word's definition. Management and the publication of a glossary is functionality is essentially the capture and publication of content. The content management system may integrate in additional functionality such as glossary term search.

4.7.9 Weblog Publication

A weblog is a collection of content pieces, called weblog entries, which are published to the World Wide Web. Weblog entries may include a variety of metadata such as tags and categorization. A weblog may be authored by one or more individuals. Weblog entries are typically displayed in chronological order. Many weblog publication systems provide the ability to filter the displayed entries by the available metadata.

Weblog publication systems have specialized forms of content syndication, such as weblog pings, weblog trackbacks, and weblog referrer tracking.

4.7.10 Bookmark Publication

Bookmark publication provides a means of publishing lists of similar items to the World Wide Web. A bookmark list consists of pieces of content, which contain a list of hyperlinks which reference content relevant to the list item.

4.7.11 Discussion Forum

A discussion forum is content publication on the World Wide Web, which contains the contributions of a variety of authors on a given topic. Each author contribution is a piece of content. Metadata associated with the captured content includes the author, a timestamp, and the topic to which the author contribution was made. The content is typically ordered chronologically or hierarchically when published.

5 Implementation Architecture, Prototype and Validation

The new functional framework is validated through implementing a pototype of the implementation architecture that demonstrates the use of a subset of the functions identified in the content management framework. The use of a portion of the framework is a practical example, as it is not required that the entire framework be used in any one content management system. This implementation is provided in order to validate and demonstrate the use of the concepts in the framework.

The prototype is a content-management-system website that captures documents in PDF and MS Word format as well as the documents' associated metadata. The user uploading a document identifies the document class and then provides the metadata for the specified class. The captured documents are processed via a simple workflow that provides approval for document publication to the website. The document text and metadata text is indexed and searchable. The search system supports queries on all of the metadata and text content or specific items of metadata.

The implementation provides user interfaces for specifying a document class hierarchy and class-specific metadata. An interface for approving documents for publication to the website is also provided.

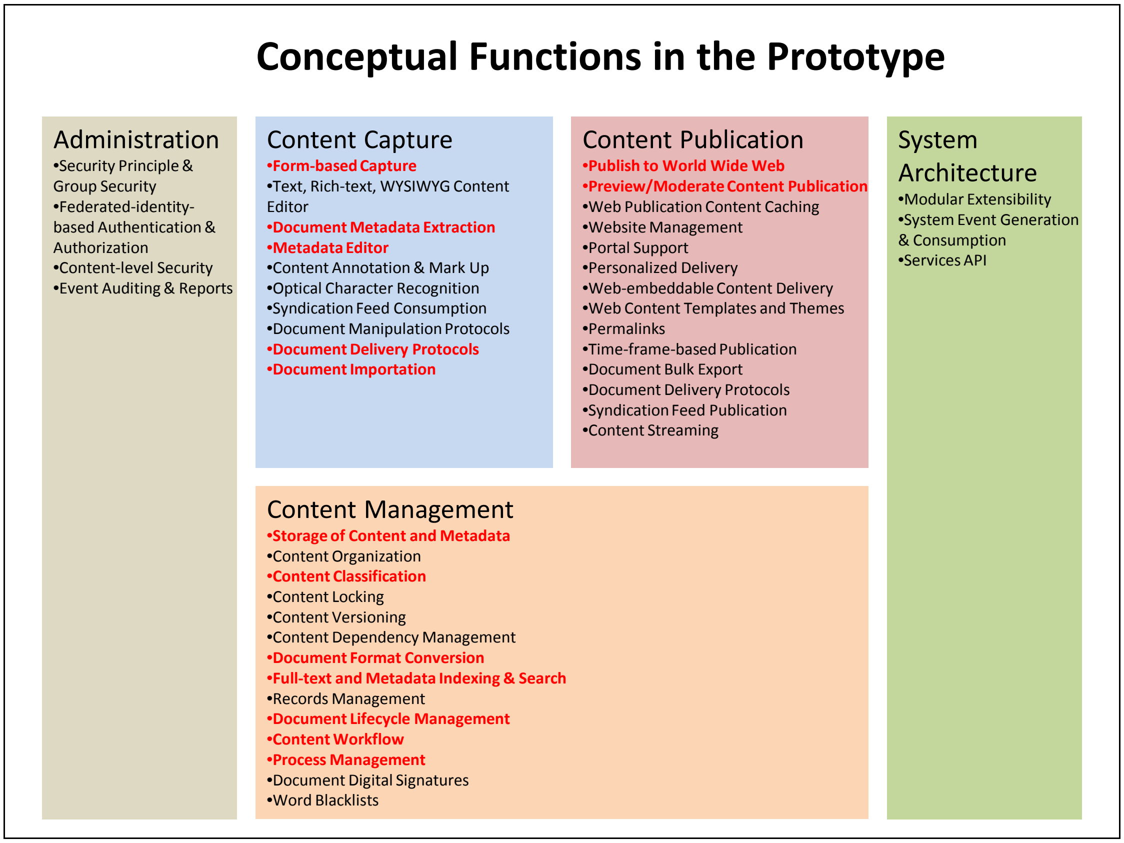
After a description of the proof-of-concept implementation and API, suggestions for future research are provided.

5.1 High Level Procedure for Identifying and Interpreting the Framework for Practical Implementation

The goal of this research is to create a new conceptual framework for the analysis existing systems and the design and development of functional modules for CMS. This framework is partially validated through the creation of prototype modules for several of the functional components, which demonstrate the use of the framework and provide an example API.

The application of the conceptual framework to the implementation architecture used in the prototype begins with the derivation of requirements from the use cases. The use cases are identified in the delimitations section of chapter 1.

For each use case, a list of conceptual functions was identified whose functionality would be necessary to implement the desired use case. The functional components, which are necessary to implement the use cases from chapter 1 are highlighted in Figure 5-1.



Conceptual Functions in the Prototype

Administration
- Security Principle & Group Security
- Federated-identity-based Authentication & Authorization
- Content-level Security
- Event Auditing & Reports

Content Capture
- **Form-based Capture**
- Text, Rich-text, WYSIWYG Content Editor
- **Document Metadata Extraction**
- **Metadata Editor**
- Content Annotation & Mark Up
- Optical Character Recognition
- Syndication Feed Consumption
- Document Manipulation Protocols
- **Document Delivery Protocols**
- **Document Importation**

Content Publication
- **Publish to World Wide Web**
- **Preview/Moderate Content Publication**
- Web Publication Content Caching
- Website Management
- Portal Support
- Personalized Delivery
- Web-embeddable Content Delivery
- Web Content Templates and Themes
- Permalinks
- Time-frame-based Publication
- Document Bulk Export
- Document Delivery Protocols
- Syndication Feed Publication
- Content Streaming

System Architecture
- Modular Extensibility
- System Event Generation & Consumption
- Services API

Content Management
- **Storage of Content and Metadata**
- Content Organization
- **Content Classification**
- Content Locking
- Content Versioning
- Content Dependency Management
- **Document Format Conversion**
- **Full-text and Metadata Indexing & Search**
- Records Management
- **Document Lifecycle Management**
- **Content Workflow**
- **Process Management**
- Document Digital Signatures
- Word Blacklists

Figure 5-1 Implemented Conceptual Functions

An API that met the required functionality for each conceptual function was designed, and the functional modules were implemented according to this specification.

The user-interface Web pages either directly or indirectly invoked the API methods for the functional interfaces as needed to perform the operations necessary to implement the desired use case.

5.2 Basic Conceptual Functions Implemented

The prototype does not include all of the basic conceptual functions of the content management framework. It does demonstrate the use of the following basic functionality:

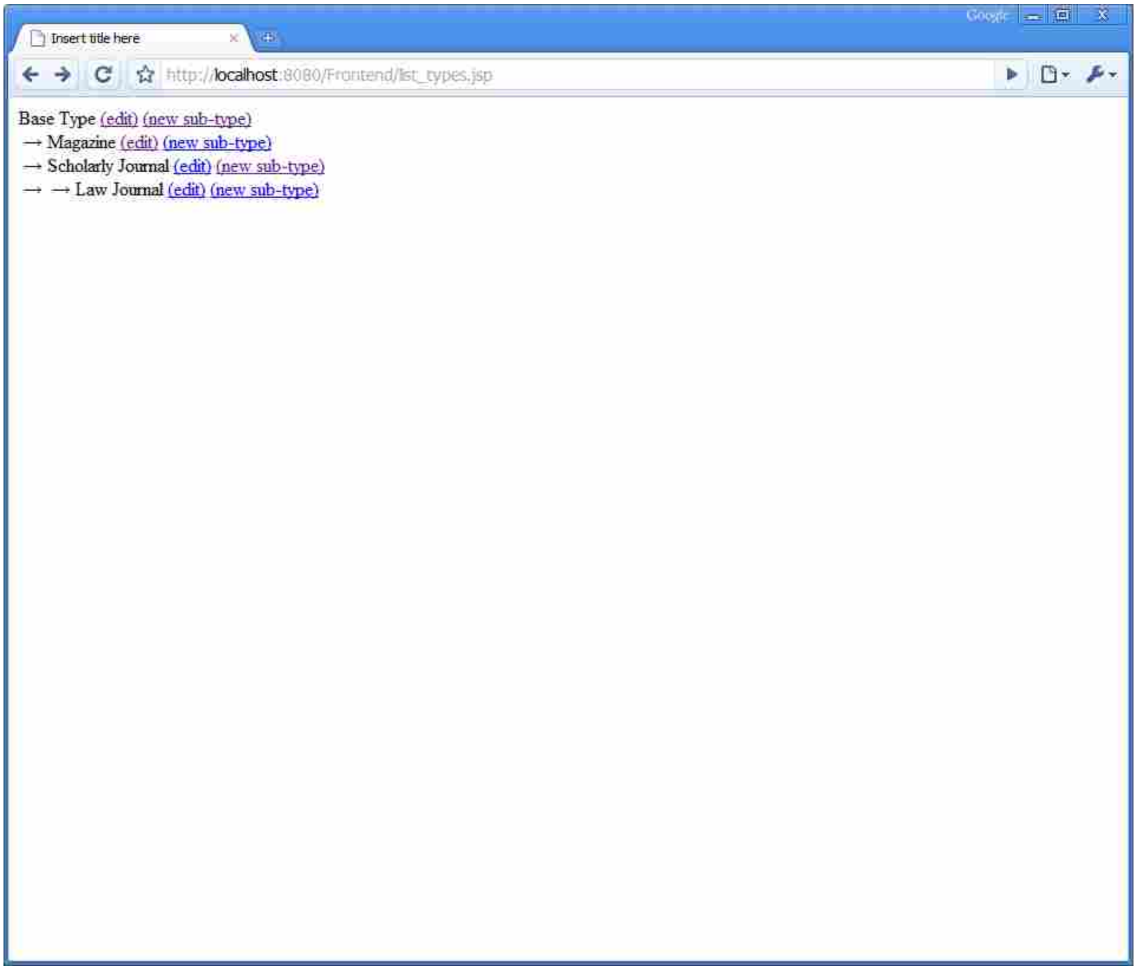
- form-based capture
- document metadata extraction
- metadata editor
- document delivery protocols
- document importation
- publication to the World Wide Web
- preview/moderate content publication
- storage of content and metadata
- content classification
- document format conversion
- full-text and metadata indexing and search
- document lifecycle management
- content workflow
- process management.

5.3 Web Interface

The prototype interacts with the users of the CMS via Web interfaces. The following describes the available Web interfaces.

5.3.1 Class Hierarchy View

This class-hierarchy-view page allows one to view the current class hierarchy. The page displays the base class at the top left. Each class's child classes are displayed as an indented list below the parent class. Next to each sub-class is the option to create a new child classes for a given class or edit the current sub-class's attributes.

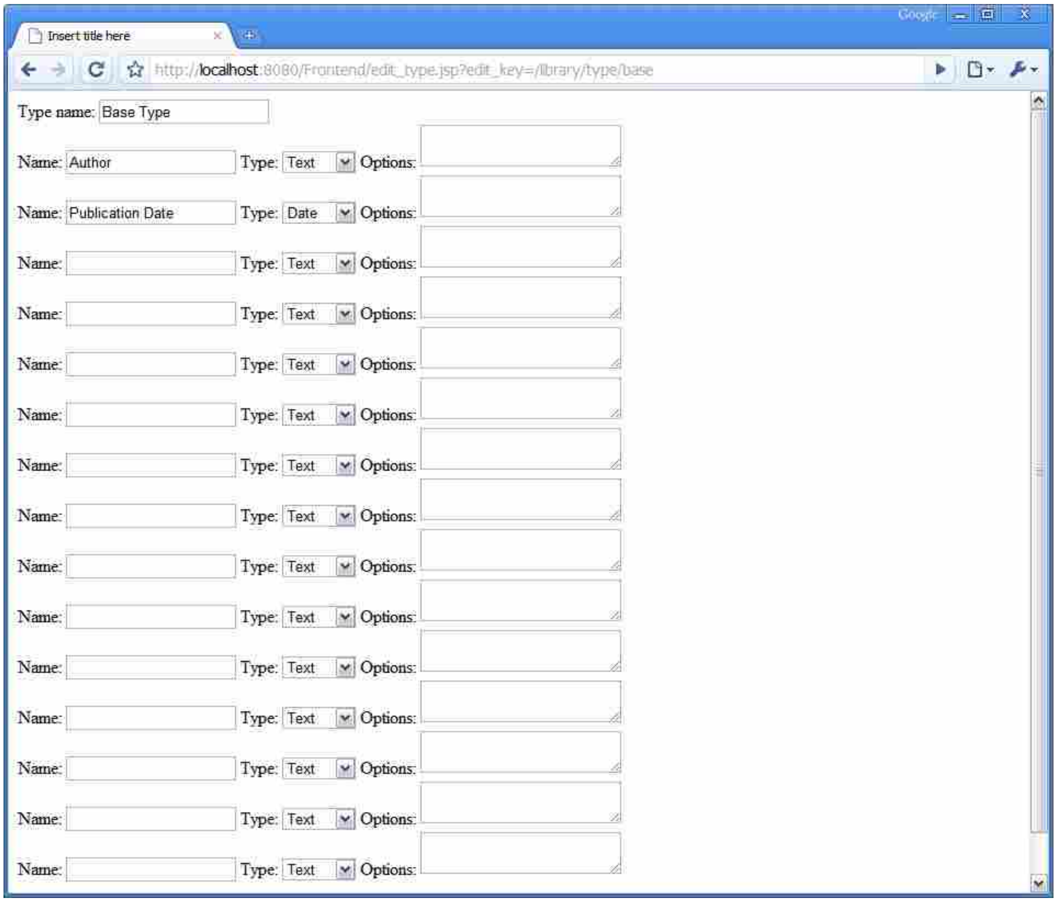


Base Type (edit) (new sub-type)
→ Magazine (edit) (new sub-type)
→ Scholarly Journal (edit) (new sub-type)
→ → Law Journal (edit) (new sub-type)

Figure 5-2 Class Hierarchy View Webpage Screenshot

5.3.2 Class Editor

The class editor allows one to edit the name and attributes of a sub-class. Sub-class attributes can be one of three data types. Supported data-types are text string, date, or a single option choice. The option-choice type allows one to identify the available options. An option is a text string.


Insert title here
http://localhost:8080/Frontend/edit_type.jsp?edit_key=/library/type/base

Type name: Base Type

Name: Author Type: Text Options:

Name: Publication Date Type: Date Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

Name: Type: Text Options:

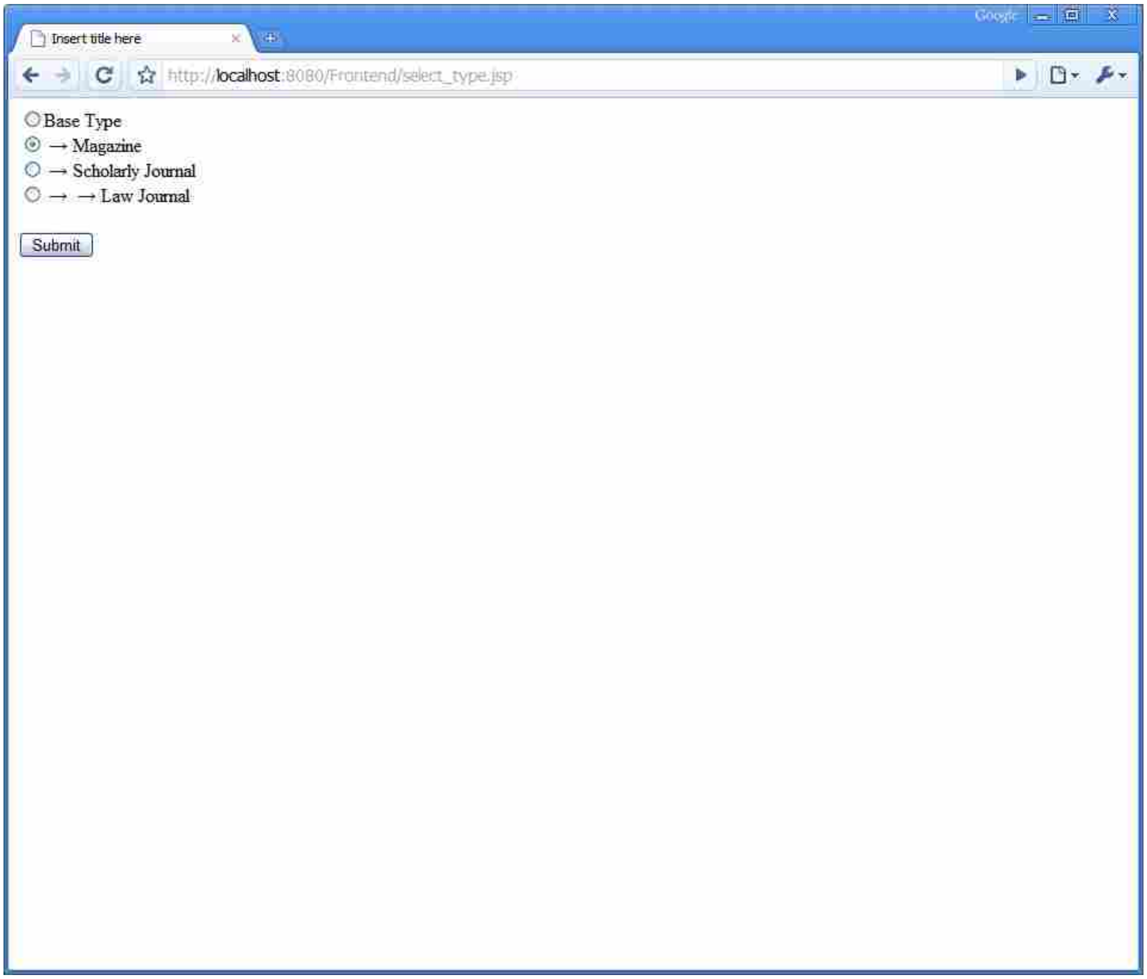
Name: Type: Text Options:

Name: Type: Text Options:

Figure 5-3 Class Editor Webpage Screenshot

5.3.3 Select Document Class

The select type screen allows one to select what sub-class best describes the document that is to be uploaded. After the sub-class is selected, a continue button causes the upload document screen to be displayed.

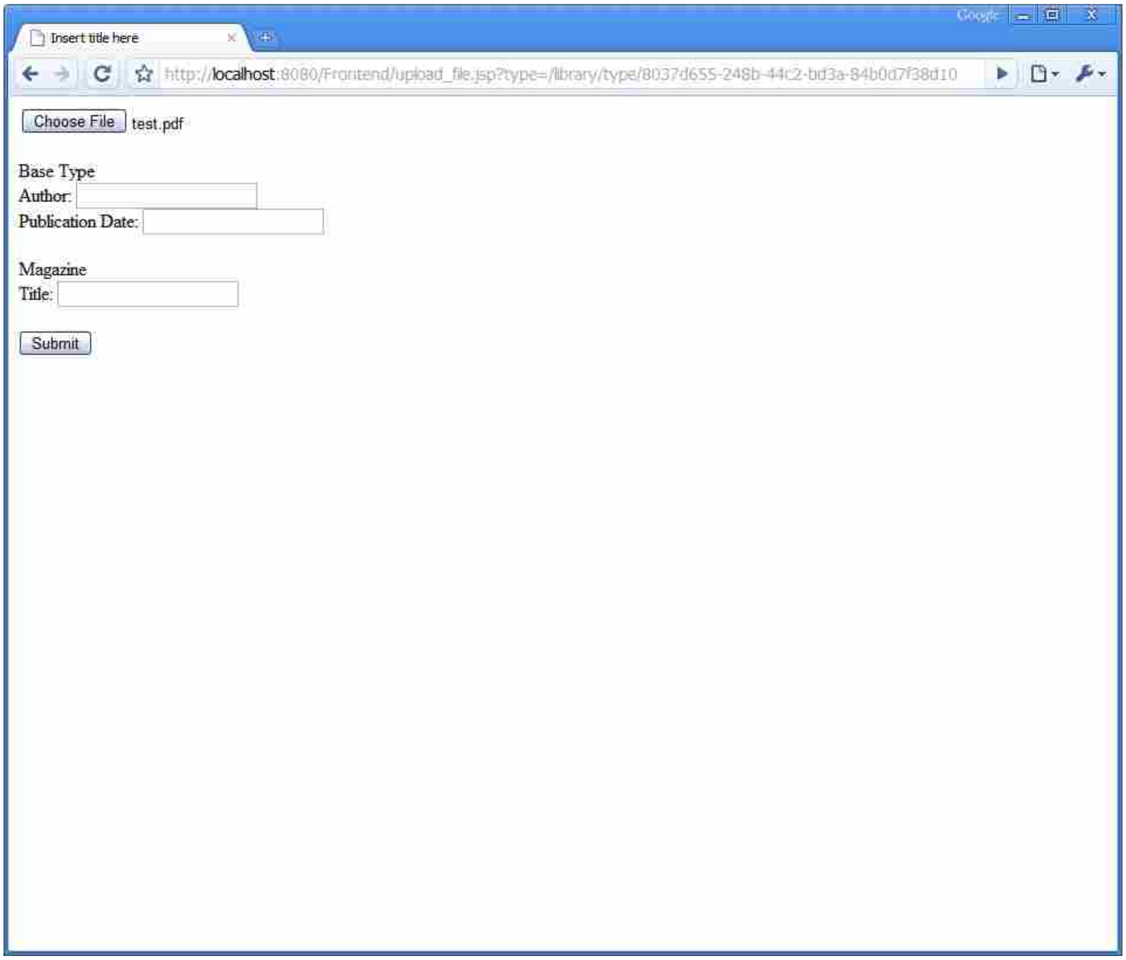


○ Base Type
● → Magazine
○ → Scholarly Journal
○ → → Law Journal

Figure 5-4 Select Document Class Webpage Screenshot

5.3.4 Upload Document

The upload document screen is displayed immediately following the select document class screen. The upload document screen allows one to select which document is to be uploaded and displays input fields for the attributes of the selected document class. Submitting the document causes the selected document to be uploaded and stored with the provided metadata.


Insert title here
Google
http://localhost:8080/Frontend/upload_file.jsp?type=/library/type/8037d655-248b-44c2-bd3a-84b0d7f38d10
Choose File
Submit

test.pdf

Base Type
Author:
Publication Date:

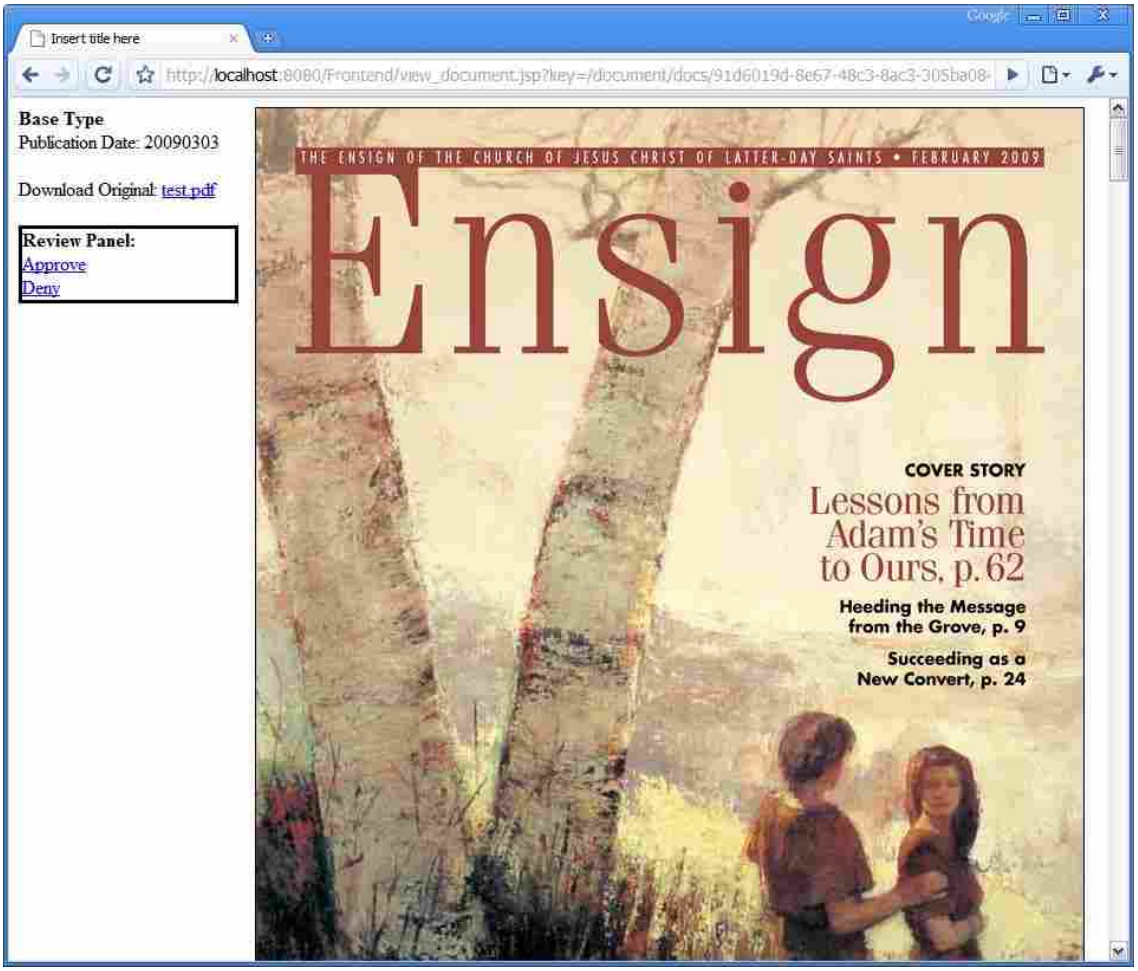
Magazine
Title:

Figure 5-5 Upload Document Webpage Screenshot

5.3.5 Approve Document

The approve document screen displays all of the documents, which have been uploaded but not yet processed by the approval workflow. A document from the unprocessed documents list is selected by the approving user, and the document is displayed. The document display screen includes a document approval pane that allows the document approver to either approve or deny the article. Based on the results of the document approval, the document is either made available for public publication on the website or the document is archived.

Approved documents can be found in the public-document search results. Documents which are denied publication and given an archived status are no longer eligible to be processed by the approval process and they do not appear as search results for public searches.


Insert title here
http://localhost:8080/Frontend/view_document.jsp?key=/document/docs/91d6019d-8e67-48c3-8ac3-305ba08-
THE ENSIGN OF THE CHURCH OF JESUS CHRIST OF LATTER-DAY SAINTS • FEBRUARY 2009
Ensign
COVER STORY
Lessons from Adam's Time to Ours, p. 62
Heeding the Message from the Grove, p. 9
Succeeding as a New Convert, p. 24

Base Type
Publication Date: 20090303

Download Original: test.pdf

Review Panel:
Approve
Deny

Figure 5-6 Approve Document Webpage Screenshot

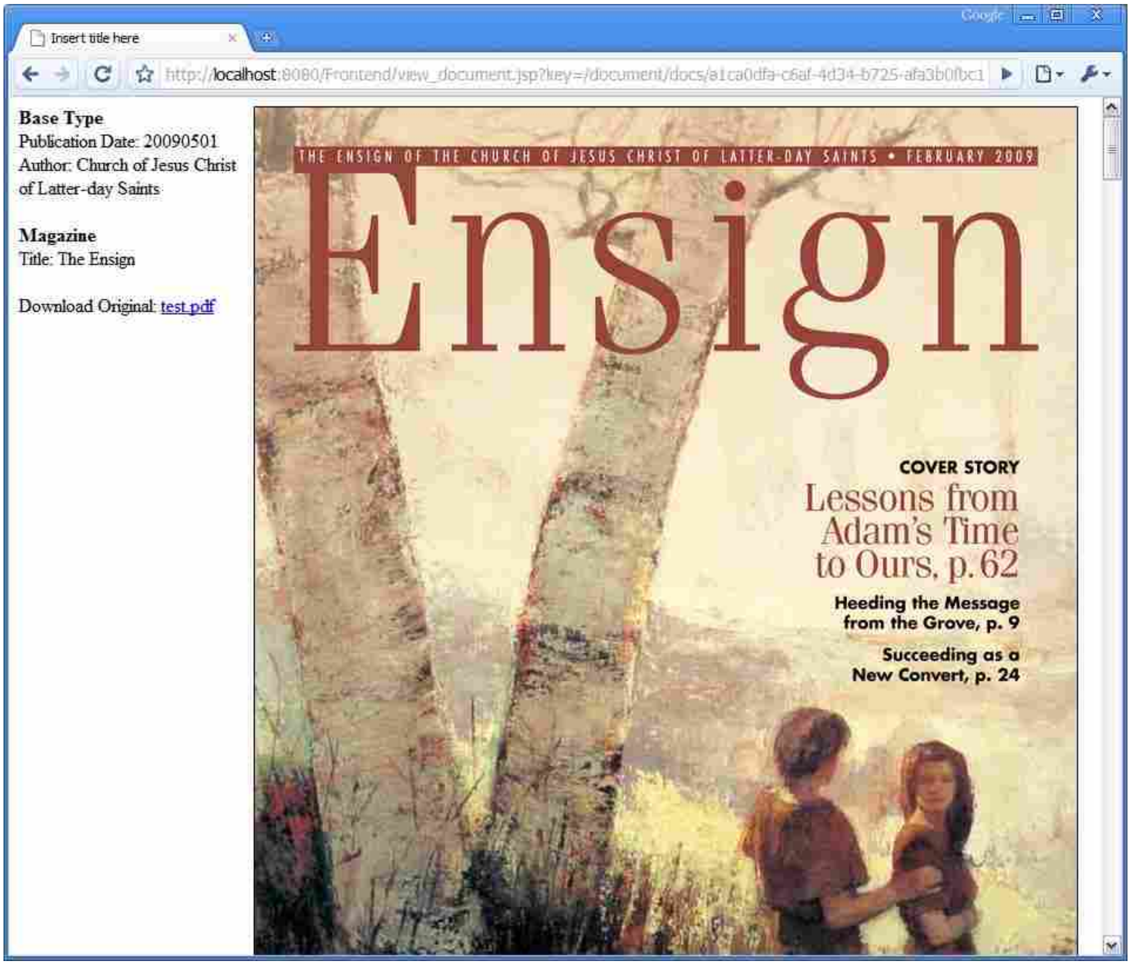
5.3.6 Display Document

The document display screen is used to display a document and its metadata. The left column of the display screen displays the document's metadata. The root document class name and metadata attributes are displayed on top of the metadata list. Each subsequent sub-class is displayed beneath the previous sub-class's metadata.

Beneath the document metadata are links that allow a visitor to the website to download the document. A link is provided for a PDF version of the document, and if the document was uploaded as a Microsoft Word document, then a link is provided for the original document.

When a document is being displayed for approval, an approval section is shown beneath the document metadata. The approval window contains a button for approving document for publication and a button for denying publication of the document.

To the right of the document metadata is shown a preview of the document. Up to the ten pages of the document are displayed as images. Images to preview the document, because they are typically faster to display than a PDF or an MS Word document, and displaying images does not require any proprietary software or plug-ins.


Insert title here
http://localhost:8080/Frontend/view_document.jsp?key=/document/docs/a1ca0dfa-c6af-4d34-b725-afa3b0fbc1
THE ENSIGN OF THE CHURCH OF JESUS CHRIST OF LATTER-DAY SAINTS • FEBRUARY 2009
Ensign
COVER STORY
Lessons from Adam's Time to Ours, p. 62
Heeding the Message from the Grove, p. 9
Succeeding as a New Convert, p. 24

Base Type
Publication Date: 20090501
Author: Church of Jesus Christ of Latter-day Saints

Magazine
Title: The Ensign

Download Original: test.pdf

Figure 5-7 Display Document Webpage Screenshot

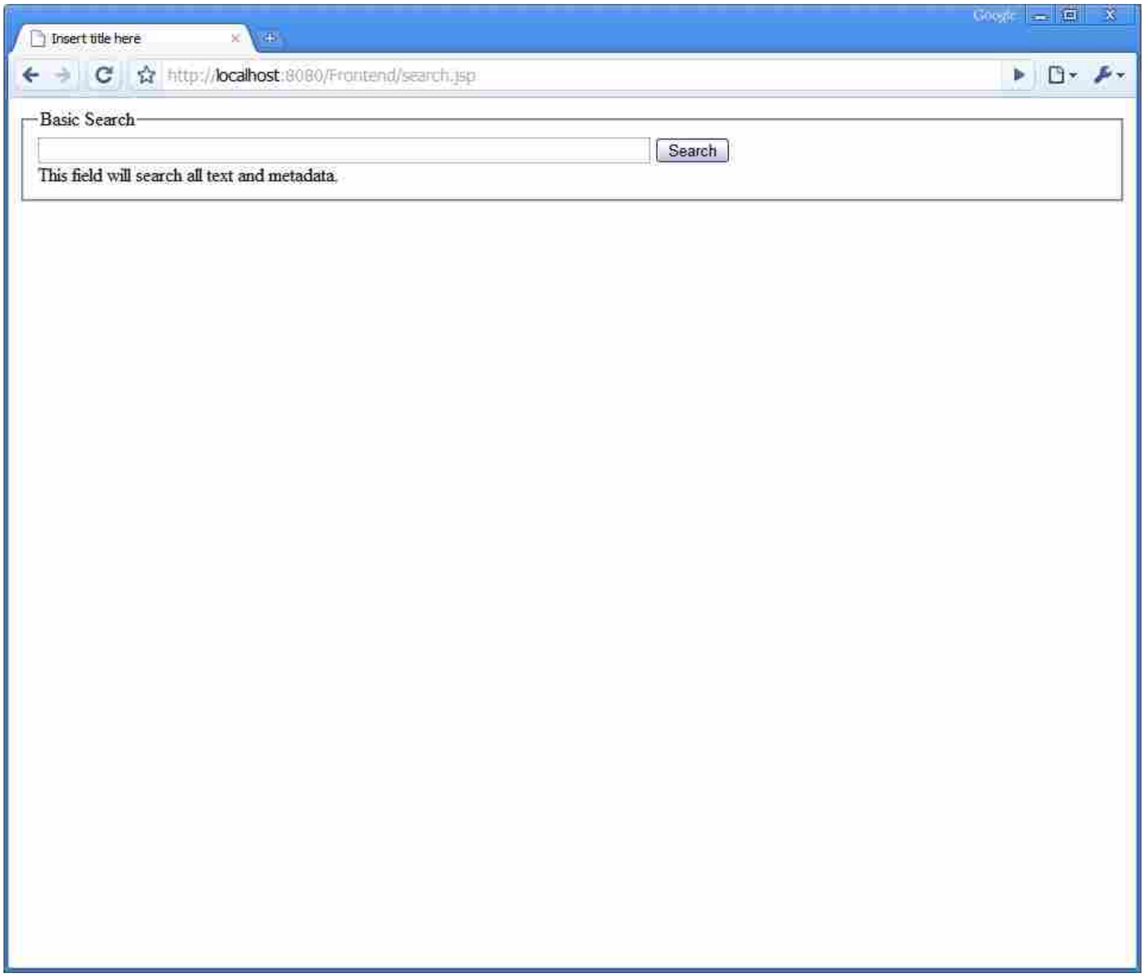
5.3.7 Process Document Console

The process document console displays the business process actions as they occur for a given documents uploaded to the website. The console is provided for demonstration purposes.

5.3.8 Search Screen

The search screen provides a text field for one to enter a search query for terms within the document text and the document metadata text. The default search strategy is to search both the document text and document metadata for documents which contain all of the terms entered into the search field.

If the appropriate metadata fields are known, documents can also be queried on based on specific metadata field values and document type.



http://localhost:8080/Frontend/search.jsp

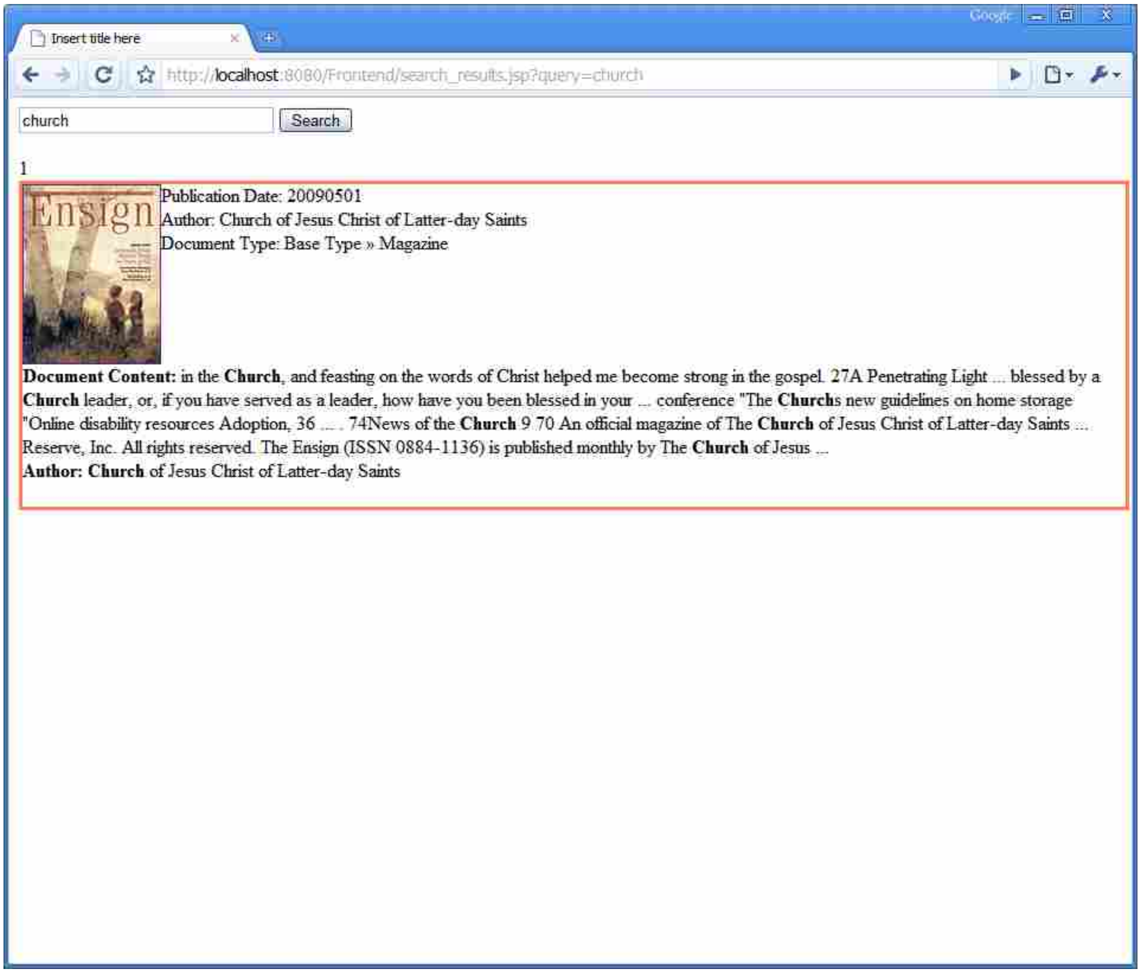
This field will search all text and metadata.

Figure 5-8 Search Screen Webpage Screenshot

5.3.9 Search Results

The search results screen displays documents which meet a given search criteria. Search queries can contain criteria based on any piece of metadata.

The search results include a thumbnail image of the first page of the document along with a summary of the relevant terms in the document.


Google
Insert title here
http://localhost:8080/Frontend/search_results.jsp?query=church
church
Search
Ensign

1

Publication Date: 20090501
Author: Church of Jesus Christ of Latter-day Saints
Document Type: Base Type » Magazine

Document Content: in the **Church**, and feasting on the words of Christ helped me become strong in the gospel. 27A Penetrating Light ... blessed by a **Church** leader, or, if you have served as a leader, how have you been blessed in your ... conference "The **Church**s new guidelines on home storage "Online disability resources Adoption, 36 ... . 74News of the **Church** 9 70 An official magazine of The **Church** of Jesus Christ of Latter-day Saints ... Reserve, Inc. All rights reserved. The Ensign (ISSN 0884-1136) is published monthly by The **Church** of Jesus ...
Author: Church of Jesus Christ of Latter-day Saints

Figure 5-9 Search Results Webpage Screenshot

5.4 Implemented Framework Components

The proof of concept implementation uses a subset of the identified components in the content management framework. The components used are the following: Document storage of content and metadata, Document lifecycle management, Search/indexing, Type management, Document format conversion, Structured data capture, File upload (file capture), Publish to web, Preview content publication, Content classification, and Content workflow.

5.4.1 Document Repository

The proof of concept implementation provides a document store that will store a data stream that represents a file and associated metadata represented as key-value pairs. The set function is used to store metadata and content. Both the key and the value are text string data types. Each document has a unique identifier called a key.

```
void setContent(String key, Map<String, String> metadata, InputStream content)
```

The key is a unique identifier for a given set of content and metadata. Prior to storing the provided metadata and content, content which exists with the same key is removed. If the content Input Stream is a null reference, then it is ignored.

```
Content getContent(String key)
```

The get operation will return a Content object which contains the metadata and content InputStream for a given key. If no Content object exists for the given key, then a null reference is returned by the operation.

5.5 Document Lifecycle Management

The documents in the proof of concept implementation have a defined lifecycle that beings with the document capture and ends in either publication or archival of the document.

The document lifecycle management interface provides a specialized view of document metadata. This document provides supports the ability to edit the current lifecycle of a given document.

```
String getLifecycle(String key)
```

This function will provide the lifecycle state of a given document identified by the provided key.

```
Set<String> getLifecycleDocuments(String lifecycle)
```

This function will provide the document keys for all of the documents with a given lifecycle.

```
void setLifecycle(String key, String lifecycle)
```

This function will set the lifecycle state of a document with a given key to the provided lifecycle state.

5.6 Document Indexing and Search

This interface provides the ability to index documents and then search for document that match a provided search query. A query defaults to searching the document content and metadata, but the query can specify specific attributes to search.

```
void index(String key, Map<String,String> metadata)
```

This method will index the provided metadata and use the key as the unique identifier.

```
List<String> search(String query)
```

This method will return the document-identifying keys whose metadata match the provided query. The query used in this implementation is a Lucene-based query.

5.7 Document Class Management

The class management interface allows one to create subtypes which, when combined into a type hierarchy consist of a document type.

```
void createSubType(String key, String parentKey, Set<Attributes> attributes)
```

This method allows one to store a subtype. The key is a unique identifier for a subtype. The parent key is the key of a given subtype's parent subtype. An attribute consists of an attribute key, an attribute name, an attribute type, and attribute options.

The supported types are text string, date, and option. When the type is an option then the attribute options consist of the possible option values.

```
SubType getSubType(String key)
```

The getSubType method returns a SubType object identified by the provided key. The object consists of the subtype key, the subtype's parent key, and the sub type attributes.

```
List<SubType> getTypeHierarchy(String key)
```

The getTypeHierarchy method returns a list consisting of the sub-type identified by the key and all of the parent sub-types. The ordered list contains the base sub-type as the first item in the list followed by each child sub-type in the type hierarchy until the identified sub-type is reached.

```
Set<SubType> getSubTypeChildren(String parentKey)
```

The getSubTypeChildren method returns all of the sub-types for the parent sub-type identified by the parent key.

5.7.1 Document Format Conversion

The following functions are used to convert document types. An alternative interface could provide a more generic method that accepted a content type string along with the file object. An additional alternative would be to accept and return other types of content references such as input stream and URIs.

```
File convertPDFToText(File pdfFile)
```

This method accepts a file object that represents a PDF. It converts the PDF file to a text file.

```
File convertDocToPDF(File docFile)
```

This method accepts a file object that represents a Microsoft Word document. It converts the Microsoft Word document to a PDF file.

```
File convertDocxToPDF(File docxFile)
```

This method accepts a file object that represents a Microsoft Word 2007 document. It converts the document to a PDF file.

```
File convertPDFPageToPNG(File pdfFile, int pageNum)
```

This method accepts a file object that represents a PDF file. The page number identifies which page of the PDF should be converted to a PNG.

```
int numPagesInPDF(File pdfFile)
```

This method accepts a file object which represents a PDF file. It determines the number of pages in the referenced PDF.

5.7.2 Resize Image

The resize image functions are used to scale the size of an image.

```
File resizeImageByHeight(File imageFile, int height)
```

This function takes a reference to an image file and scales the image so that the height of the image is the provided height in pixels. The function returns a reference to the newly scaled image. Images are scaled proportionally.

```
File resizeImageByWidth(File imageFile, int width)
```

This function takes a reference to an image file and scales the image so that the width of the image is the provided width in pixels. The function returns a reference to the newly scaled image. Images are scaled proportionally.

```
File resizeImage(File imageFile, int height, int width)
```

This function accepts a reference to an image file and scales the image so that the height or the width of the image is either the height or width in pixels of the provided width or height. The image will be scaled so that neither the height nor the width will exceed the provided measurements. The function returns a reference to the newly scaled image. Images are scaled proportionally.

5.7.3 Structured Data Capture

The proof of concept implementation provides a web-based form generation component which takes accepts a sub-type key as a component and displays the entire type, sub-type names and attributes. A text field is provided for date and text entry. For option attributes, the option attributes are displayed as possible values for a select element on the web page.

When submitted, the form data is captured and stored by a Java-based Web application.

5.7.4 File Upload

A web-based software component is used in this proof-of-concept implementation which accepts a form-based file upload. When the form is submitted, the uploaded file is stored by the system using a Java-based Web application.

5.7.5 Publish to Web

The proof-of-concept implementation provides a publish-to-web software component, which accepts a document key as an argument and generates a webpage. The webpage consists of the document metadata, an image-based preview of the document, and links to the PDF-formatted document, if available, and the originally uploaded document.

5.7.6 Preview Content Publication

The proof of concept implementation provides a means of viewing documents, which have not been approved for public publication. This allows a document approver to preview the documents before the system makes them publically available.

5.7.7 Content Classification

The proof of concept implementation provides a means of classifying documents being captured by the system. Based on the selected classification, the applicable document type's metadata can be captured and displayed on demand. The implementation stores the metadata along with descriptive data about the document type and its attributes.

5.7.8 Content Workflow

The proof of concept implementation supports a basic workflow for document processing, approval, and publication. Captured documents and metadata are initially processed and then the document life cycle indicates that these documents are pending approval. A document approver can view the documents needing approval and determine whether they are ready for publication or should be archived. Documents approved for publication are then made available via the systems public document search functionality.

In this case, the workflow is based on the document lifecycle, so the document lifecycle interface is used to manage the workflow.

5.8 Document Renditions

A captured document is stored as multiple content objects within the document store. A document typically exists as a document object and a number of rendition objects. The document objects consist of the of the document metadata. The document object metadata points to additional objects, which are rendition objects.

A rendition object consists of the binary document data and a document content type. In this implementation, rendition objects are the originally uploaded document, and documents that were generated from the original document, such as a PDF version of a Microsoft Word document or an image generated from a PDF document.

Rendition objects are treated as immutable objects by the implementation, while the document object containing the document metadata has metadata which is modified throughout the document lifecycle.

5.9 Document Metadata

The documents in the proof of concept implementation have a variety of metadata associated with them. Some metadata is universal to all documents in the system, while other pieces of metadata are specific to the document type.

In the case of this proof-of-concept implementation, all metadata is stored as key-value pairs. Both the key and the value are stored as strings and associated with the document. Due to the non-specific nature of the metadata store, metadata key selection is important to managing the metadata. Specification of the metadata value is also important. For example, some metadata items represent multiple values as a single metadata item by tab-delimiting the items within the value string.

The metadata value string does not have a specified limit. It is, however, limited by the underlying implementation.

5.9.1 Document Universal Metadata

Universal metadata represents attributes common to all documents, such as when the document was captured, the document lifecycle state, the document text content, etc. The universal metadata stored in this content management system implementation is shown in Table 5-1. Document universal metadata is stored using key-value pairs.

Table 5-1 Document Universal Metadata

Metadata	**Description**
Key	The key is a unique identifier for this object in the database.
Life Cycle State	The life cycle state tracks the current state of this document object in its life cycle.
Capture Date	The capture date identifies the date that his document was uploaded.
Processing Complete Date	This is a timestamp for when the post-capture processing of this document was complete.
Approval Date	The approval date is the date when the approval identified this document as being approved for publication or identified as being an archive only document.
Approving User UUID	This is the UUID for the user who submitted the approval of this document object.
Original Rendition Key	This is the unique identifier for the rendition object which represents the document that was originally uploaded.
PDF Rendition Key	This is the key for the rendition object that represents the PDF rendition of the originally uploaded document. If the originally uploaded document was a PDF, then the key will be the same as the original rendition key.
Text Content	The text content metadata is a character string of the textual content of the document.
Submitter User UUID	This is the UUID of the user who submitted the document.
Type Hierarchy Key List	This item of metadata is a list of keys which represent the sub-types that make up the document type. The root sub-type is the first key in the list followed by subsequent child sub-types. The keys are tab separated.
Type Hierarchy Name List	This item of metadata is a list of the names of the sub-types that make up the document type. The root sub-type is the first name in the list followed by subsequent child sub-types. The type names are tab separated.
Image Rendition Key List	This item of metadata contains an ordered list of keys that reference rendition objects which contain images of each page of the document. The first document is the first key referenced, etc. The keys are tab separated.
Document Thumbnail Image	The document thumbnail image metadata contains a key that references a rendition object which contains a thumbnail-size rendition of the first page of the document.
Attribute UUID List	This metadata contains a list of the UUIDs for the metadata attributes for this object. Each attribute on each sub-type has its own UUID. The list of UUIDs is tab separated.

5.9.2 Document Attribute Metadata

The type-specific metadata is dependent on how a document is classified. Due to the varying nature of the attributes for a given type, the metadata stored with the document will vary. The attribute values as well as the attribute names and types are stored as metadata with the document. The metadata descriptive of the document attributes is stored, so that there is not a dependence on the document type repository for this information.

A document may include one or more attributes. Each document attribute has a name, a value, a type, and a UUID associated with it. Based on the attribute UUID list in the previous metadata item, each attribute name, value, and type can be found in the included metadata. This information is stored as metadata by identifying the components of the attribute with a metadata key that begins with either “attribute-name-“, “attribute-value-“, or “attribute-type-key-“ followed by the attribute UUID.

5.9.3 Rendition Metadata

The rendition metadata is universal to all rendition objects. A rendition object consists of a rendition key, the document content, and the document content type.

The document content type is the mime-type of the document. The rendition key is the rendition object’s unique identifier. Finally the document content is a data stream that is stored by the content management system.

5.10 Document Format Detection

The content management system determines the document format based on the document filename extension. Files ending in “.doc” are assumed to be Microsoft-Word-2003-formatted documents, and so on. Currently, the supported document formats are PDF, and Microsoft Word 2003 and 2007 formats.

5.11 Implementation Details

The content management system is implemented primarily in Java 5 (Sun Microsystems, Inc. "Java SE Overview: At a Glance.", n.d.). Web pages are generated and processed using the Apache Tomcat 6.0
(The Apache Software Foundation. "Apache Tomcat." 2009). The text indexing and search query system is implemented using the Java version of Apache Lucene version 2.4 (The Apache Software Foundation. “Welcome to Lucene!” 2009). Persistence to a relational database is enabled using the Hibernate ORM Persistence Library (Redhat Middleware, LLC, n.d.).

Metadata and content data are stored using the MySQL Database version 5.0 (Sun Microsystems, Inc. "MySQL: The World's Most Popular Open Source Database.", n.d.). Scratch files for document conversion are stored using the local file system.

Image resizing is performed using ImageMagick 6.5 (imageMagick Studio, LLC., n.d.). The number of pages in a PDF is also determined using ImageMagick.

Microsoft Word documents in version 2003 and version 2007 formats are converted to PDF using OpenOffice.org (OpenOffice.org, n.d.). An OpenOffice.org macro is used to invoke this functionality.

A page of a PDF document is converted to an image using the command line version of Ghostscript 8.63 (Ghostscript 2008).

Functionality by applications external to the Java Runtime is invoked via the Java Runtime Exec method.

This implementation does not provide support for many common use cases, such as user management or security.

5.12 Successful Utilization of the New Conceptual Framework

The prototype successfully implemented a subset of the conceptual functions from the functional framework. The prototype used the functional modules to satisfy the requirements of the use cases from chapter 1.

6 Conclusions and Recommendations

6.1 Conclusions

This research has demonstrated the utility of a new functional framework for content management. The framework was successfully used to prototype an implementation architecture that satisfies the requirements of the set of use cases from chapter 1.

6.1.1 New Framework Overcomes Problems with Existing CMS Models

The new framework provides a foundation for the research, analysis, development and discussion of CMSs. The prototype provides an example of how the new framework can be used to alleviate inconsistent communication and confusion regarding CMS functionality while providing a practical means of identifying component types and functional modules as a foundation for CMS development.

See chapter 4 and 5 for a more in-depth discussion of the supported component types and functional modules of the CMS framework for content management.

6.1.2 Successful use of the New Framework

The functional framework for content management worked well as a tool for identifying the functional boundaries for the implementation requirements. The application of the conceptual concepts of the framework to a working prototype was also successful. The procedure of identifying use cases, determining the applicable functional components, and then implementing these components to support a use case has proven to be a valid use of the framework.

This research is limited in that the new framework has only been applied to a single proof-of-concept implementation, which does not include all of the functional components. Additional implementations of the functional modules and their application to more use cases will be useful in further validating the framework and confirming its usefulness in practical implementations.

6.2 Recommendations for Future Research

6.2.1 Additional Framework Component Implementations

Additional implementations of conceptual functions will further refine and validate the functional framework. This will identify additional common use cases, which will aide in an understanding of the model. Of particular interest is the area of document security, such as the use of encryption for the validation and protection of data.

Additionally, alternative implementations of the functional modules will aide in understanding of the framework by identifying alternative ways of providing the desired functionality. Alternate implementations include the use of the functional modules as integrated and consolidated software systems or alternatively as distributed software systems. These different software architecture strategies may cause significant differences in the APIs. Both, identifying common use cases and alternative means of implementing functionality aide in the overall goal of understanding content management systems and the continued development of these tools.

7 Bibliography

Alkacon Software GmbH. "Download the OpenCms Documentation." http://opencms.org/en/download/documentation.html (accessed March 21, 2009).

Alkacon Software, GmbH. "OpenCms: Professional Content Management." OpenCms. http://www.opencms.org/en/ (accessed on January 3, 2009).

Alfresco Software, Inc. "Open Source Enterprise Content Management System (CMS) by Alfresco." http://www.alfresco.com/ (accessed on January 3, 2009).

Alfresco Software, Inc. "About Alfresco - The Open Source Alternative for Enterprise Content Management." http://www.alfresco.com/about/ (accessed on March 21, 2009).

Alfresco Software, Inc. "Alfresco Developers." http://wiki.alfresco.com/wiki/Main_Page (accessed on May 27, 2008).

Apache Lenya - Open Source Content Management (Java/XML). The Apache Software Foundation, http://lenya.apache.org/ (accessed on January 3, 2009).

Apache Software Foundation, The. "Apache Tomcat." Apache Tomcat. 2009, http://tomcat.apache.org/ (accessed March 21, 2009).

Apache Software Foundation, The. "Roller." *Apache Roller: Open Source Java Blog Server.* http://roller.apache.org/ (accessed January 3, 2008).

Apache Software Foundation, The. "Roller 3.1 User Guide." April 2007, http://www.apache.org/dist/roller/roller-3/v3.1.0/docs/roller-user-guide.pdf (accessed on May 27, 2008).

Apache Software Foundation, The. "Welcome to Lucene!" Apache Lucene. 2009, http://lucene.apache.org/ (accessed March 21, 2009).

APM Group Ltd, The. "ITL Home." March 3, 2009, http://www.itil-officialsite.com/home/home.asp (accessed March 21, 2009).

Brown, A. and D. Maiworm. "Symantec Enterprise Vault Technical White Paper: File System Archiving." Symantec, Inc. August 27, 2007, http://www.datanetworks.com/WhitePapers/Symantec%20-%20File%20System%20Archiving.pdf (accessed March 21, 2009).

Bruillard, E. and G. L. Baron. "Computer-based Concept Mapping: A Review of a Cognitive Tool for Students." *Proceedings of Conference on Educational Uses of Information and Communication Technologies* (ICEUT 2000): 331-338.

Bush, V. "As We May Think." *The Atlantic*, July 1945, http://www.theatlantic.com/doc/print/194507/bush (accessed January 3, 2009).

Cofax: Content Object Factory. http://www.cofax.org (accessed on January 3, 2009).

Consortium Caroline. Claroline.NET http://www.claroline.net/ (accessed on January 3, 2009).Moodle. http://moodle.org/ (accessed on January 3, 2009).

Doculabs. "Documentum's Fit in an Enterprise Reference Architecture for Content Management." 2004. http://www.documentum.ru/pdf/Reviews/doculabs_era_1204.pdf (accessed April 29, 2009).

Drupal. http://drupal.org/ (accessed on January 3, 2009).

Drupal: Getting Started. http://drupal.org/getting-started (accessed on January 3, 2009).

Drupal.org. "Drupal Handbooks." 2009, http://drupal.org/handbooks (accessed March 21, 2009).

Drupal.org. "History: About the Drupal Project." January 16, 2009, http://drupal.org/node/769 (accessed March 21, 2009).

DSpace. http://www.dspace.org/ (accessed on January 3, 2009).

EPrints Free Software. EPrints. http://www.eprints.org/software/ (accessed on January 3, 2009).

Fedora Commons. “Fedora Commons: Home”. http://www.fedora.info/ (accessed on January 3, 2009).

Ghostscript. "Ghostscript, Ghostview, and GSview." 2008, http://pages.cs.wisc.edu/~ghost/ (accessed March 21, 2009).

Gingell, D. "A 15 Minute Guide to Enterprise Content Management." EMC Corporation, March 2006, http://www.aiim.org.uk/publications/ecm_at_work/pdfs/Ecm_15min_guide.pdf (accessed January 3, 2009).

Han, Y. "Digital Content Management: The Search for a Content Management System." *Digital Content Management: The Search for a Content Management System* 22, no. 4 (2004): 355-365, http://www.emeraldinsight.com/10.1108/07378830410570467 (accessed on May 2, 2008).

Higgins, B. "This blog is now on Roller, and I have no idea how to use it." *developerWorks: Jazz Platform Development.* Entry posted March 12, 2006. http://www.ibm.com/developerworks/blogs/page/BillHiggins?entry=this_blog_is_now_on (accessed March 21, 2009).

ImageMagick Studio, LLC. "ImageMagick: Convert, Edit, and Compose Images." http://www.imagemagick.org/script.index.php (accessed March 21, 2009).

Information Systems Audit and Control Association. "COBIT 4.1." 2009, http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=31519 (accessed March 21, 2009).

Johnson, D. "A Not So Brief History of Roller." *Blogging Roller: Dave Johnson on Blogging, Open Source and Java.* Entry posted December 31, 2002. http://rollerweblogger.org/roller/date/20021231 (accessed January 3, 2009).

Maedche, A., and others. "Ontologies for Enterprise Knowledge Management." *IEEE Intelligent Systems* 18, no. 2 (March 2003): 26-33, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1193654&isnumber=26843 (accessed April 14, 2008).

Mambo Foundation, Inc. Mambo. http://mambo-foundation.org/ (accessed on January 3, 2009).

Martin, D. 2004. Moodle: A Virtual Learning Environment for the Rest of Us. *Teaching English as a Second or Foreign Language* 8, no. 2 (September): under “M-2.” http://tesl-ej.org/ej30/m2.html (accessed March 21, 2009).

McGrath, J."Open-source CMS: On The Rise." *ZDNet Australia.* November 28, 2002, http://www.zdnet.com.au/news/business/soa/Open-source-CMS-On-the-rise/0,139023166,120270243,00.htm (accessed March 21, 2009).

McNay, H. E. "Enterprise Content Management: An Overview." *Proceedings of IEEE International Professional Communication Conference* (2002): 396-402, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1049123&isnumber=22469 (accessed January 3, 2009).

Midgard Project, The. Midgard: Open Source Content Management Framework. http://www.midgard-project.org/ (accessed on January 3, 2009).

MoodleDocs."Moodle Docs: Main Page." http://docs.moodle.org/en/Main_Page (accessed on January 3, 2009).

Moore, C. and R. Markham. "Enterprise Content Management: A Comprehensive Approach for Managing Unstructured Content." Giga Information Group, Inc., April 5, 2002, http://www.msiinet.com/html/pdfs/essecm3.pdf (accessed April 14, 2008).

Mukherjee, R. and J. Mao. "Enterprise Search: Tough Stuff." *ACM Queue* 2, no. 2 (April 2004): 36-46, http://doi.acm.org/10.1145/988392.988406 (accessed January 3, 2008).

Munkvold, B. E. and others. "Contemporary Issues of Enterprise Content Management: The Case of Statoil." *Scandinavian Journal of Information Systems* 18, no. 2

(January 2006), http://www.e-sjis.org/journal/volumes/volume18/no2/munkvoldetal-18-2.pdf (accessed on January 3, 2009).

New Zealand Digital Library Project. Greenstone Digital Library Software. http://www.greenstone.org/ (accessed on January 3, 2009).

OpenOffice.org. "OpenOffice.org: The Free and Open Productivity Suite." http://www.openoffice.org/ (accessed March 21, 2009).

Pemberton, S. and J. Boyer. "The Forms Working Group." W3C Forms Working Group. March 5, 2009, http://www.w3.org/MarkUp/Forms/ (accessed March 21, 2009).

PHPNuke.org. PHP-Nuke. http://phpnuke.org/ (accessed on January 3, 2009).

Plone Foundation and others, The. "Plone CMS: Open Source Content Management." http://plone.org/ (accessed on January 3, 2009).

Plone Foundation and others, The. "Plone Manuals." http://plone.org/documentation/manual (accessed on January 3, 2009).

phpCollab. http://www.php-collab.com/blog/ (accessed on January 3, 2009).

Public Knowledge Project. Open Journal Systems. http://pkp.sfu.ca/?q=ojs (accessed on January 3, 2009).

Redhat Middleware, LLC. "Relational Persistence for Java and .NET." www.hibernate.org. http://www.hibernate.org/ (accessed March 21, 2009).

Reimer, J. A. "Enterprise Content Management." *DatenBank-Spektrum (*April 2002): 17-22, http://www.dbis.prakinf.tu-ilmenau.de/papers/dbspektrum/dbs-04-17.pdf (accessed on January 3, 2009).

SpringCM. "SpringCM: On-demand Document Management and Workflow." http://www.springcm.com/products (accessed on January 3, 2009).

Stylite GmbH. eGroupWare. http://www.egroupware.org/ (accessed on January 3, 2009).

Sun Microsystems, Inc. "Java SE Overview: At a Glance." http://java.sun.com/javase/ (accessed March 21, 2009).

Sun Microsystems, Inc. "MySQL: The World's Most Popular Open Source Database." www.mysql.com. http://mysql.com/ (accessed March 21, 2009).

Symantec Corporation."Symantec Enterprise Vault." 2009, http://www.symantec.com/business/products/overview.jsp?pcid=pcat_storage&pvid=322_1 (Accessed January 3, 2009).

Symantec Corporation. Symantec Enterprise Vault 7.0. http://www.symantec.com/business/products/overview.jsp?pcid=pcat_storage&pvid=322_1 (accessed on May 27, 2008).

Tanczos, M. "Moodle: An Open Source Learning Management System for Hybrid Learning in Secondary-level Schools." *Moodle.* September 10,2006, http://www.easdonline.com/mod/resource/view.php?id=345 (accessed March 21, 2009).

Terra, J. C. and T. Angeloni. "Understanding the Difference between Information Management and Knowledge Management." Terraforum Website, 2002, http://www.kmadvantage.com/docs/kmarticles/Understanding_the_Difference_between_IM_and_KM.pdf (accessed on January 3, 2009).

Typo3 Association. TYPO3 CMS. http://typo3.com/ (accessed on January 3, 2009).

Tramullas, J. "Open Source Tools for Content Management." *Hipertext.net*, no. 3 (2005), http://www.hipertext.net/english/pag1013.htm (accessed January 3, 2009).

Tyrvainen, P., A. Salminen, and T. Paivarinta. "Introduction to the Enterprise Content Management Minitrack." *Proceedings of the 36th Annual Hawaii International Conference on System Sciences* (January 6-9, 2003): 104, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1174244&isnumber=26341 (accessed on April 14, 2008).

Tyrväinen, P., and others. "Characterizing the Evolving Research on Enterprise Content Management." *European Journal of Information* 14, no. 6 (December 2006): 627-634, https://www.lib.byu.edu/cgi-bin/remoteauth.pl?url=http://proquest.umi.com/pqdweb?did=1192815161&Fmt=2&clientId=9338&RQT=309&VName=PQD (accessed April 14, 2008).

U.S. Department of Commerce. "NIST Special Publication 800-53 Revision 2: Recommended Security Controls for Federal Information Systems." December 2007, http://csrc.nist.gov/publications/nistpubs/800-53-Rev2/sp800-53-rev2-final.pdf (accessed March 21, 2009).

U.S. Department of Defense. "DoD 5015.02-STD: Electronic Records Management Software Applications Design Criteria Standard." April 25, 2007, http://www.dtic.mil/whs/directives/corres/html/501502std.htm (accessed on March 21, 2009).

U.S. Department of Labor: Occupational Safety and Health Administration. http://www.osha.gov/ (accessed on January 3, 2009).

U.S. Department of State Freedom of Information Act. http://www.state.gov/m/a/ips/ (accessed on January 3, 2009). U.S. Public Law 107-204. "Sarbanes-Oxley Act of 2002." July 30, 2002, http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ204.107 (accessed on January 3, 2009).

U.S. Environmental Protection Agency. http://www.epa.gov/ (accessed on January 3, 2009).

U.S. Food and Drug Administration. "FDA: Title 21 Code of Federal Regulations (21 CFR Part 11) Electronic Records; Electronic Signatures." FDA Office of Regulatory Affairs, March 2003, http://www.fda.gov/ora/compliance_ref/Part11/ (accessed on January 3, 2009).

U.S. Securities and Exchange Commission. "SEC: 17 CFR PARTs 240 and 242." November 2, 2001, http://www.sec.gov/rules/final/34-44992.htm (accessed on January 3, 2009).

Vyas, M. and others. "Sun Blogs: A Sun Java System Web Server 7.0 Reference Deployment." *BigAdmin System Administration Portal.* October 2008, http://www.sun.com/bigadmin/features/articles/sunblogs_webserver.jsp (accessed March 21, 2009).

Weiseth, P. E. and others. "The Wheel of Collaboration Tools: A Typology for Analysis Within a Holistic Framework." *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work* (2006): 239-248, http://doi.acm.org/10.1145/1180875.1180913 (accessed on January 3, 2009).

Weitzman, L. "Meta-design for Sensible Information." *ACM Interactions* 11, no. 2 (March 2004): 71-73, http://doi.acm.org/10.1145/971258.971284 (accessed January 3, 2009).

Weitzman, L. and others. "Transforming the Content Management Process at IBM.com." *Conference on Human Factors in Computing Systems* (2002): 1-15, http://portal.acm.org/ft_gateway.cfm?id=507757&type=pdf&coll=GUIDE&dl=GUIDE&CFID=17054450&CFTOKEN=19207502 (accessed on January 3, 2009).

WikiMedia Project, A. MediaWiki. http://www.mediawiki.org/wiki/MediaWiki (accessed on January 3, 2009).

Wolf, G. "The Curse of Xanadu." *Wired* 3, no. 6 (June 1995), http://www.wired.com/wired/archive//3.06/xanadu_pr.html (accessed January 3, 2009).

WordPress. "Word Press > Blog Tool and Publishing Platform." http://wordpress.org/ (accessed on January 3, 2009).

Yan, N., D. Leip, and K. Gupta. "The Use of Open-source Software in the IBM Corporate Portal." *IBM Systems Journal* 44, no. 2 (2005): 419-425, http://portal.acm.org/ft_gateway.cfm?id=507757&type=pdf&coll=GUIDE&dl=GUIDE&CFID=17054450&CFTOKEN=19207502 (accessed on January 3, 2009).

Yin, Y. and others. "Comparison of two concept-mapping techniques: Implications for scoring, interpretation, and use." *Journal of Research in Science Teaching* 2, no. 42 (2005): 166-184.

Zope Community. Zope. http://www.zope.org/ (accessed on January 3, 2009).

Zykov, S. V. "Enterprise Content Management: Theory and Engineering for Entire Lifecycle Support." *Proceedings of the 8th International Workshop on Computer Science and Information Technologies* 1 (September 28, 2006): 92-97, http://arxiv.org/ftp/cs/papers/0607/0607122.pdf (accessed on January 3, 2009).

APPENDICES

Appendix A
Content Management System Functional Analysis

Component Type	Functional Component	Description	Alresco ECM	Apache Roller	Drupal	Moodle	OpenCms	Plone	SpringCM	Enterprise Vault
Web Content Management	Document publication to web	Support the publication of documents to the web	•	•	•	•	•	•	•	
Web Content Management	Website management	The publication and management of non-specialized web sites	•		•		•	•		
Web Content Management	Web content change auditing	Track changes to web-published content	•							
Web Content Management	Site snap-shots	Track changes by entire website	•							
Web Content Management	Integrated records management	Integrated records management functionality for tracking web content	•							
Web Content Management	Portal/portlet support	Support portal integration standards	•					•		
Web Content Management	Contextual content delivery	Personalization based on requesting user preferences and user metadata	•							
Web Content Management	Content dependency management	Verify that references between web-published content are valid	•				•			

Web Content Management	Form input (Xforms)	Support Xforms standards for form descriptions	•							
Web Content Management	Embeddable content support	Support the embedding of content in other websites	•							
Web Content Management	Transactional content deployment	Publish entire sets of content as a single transactional change	•							
Web Content Management	Content templates	Support for the template-based look and feel of web content	•		•		•	•		
Web Content Management	Multi-channel publishing (XHTML, PDF etc)	Support for publication of content to the web in a variety of formats	•						•	
Web Content Management	Website content versioning	Support for the tracking of multiple versions of web-published content	•							
Web Content Management	Structured content (book) publication (multi-page)	Publication of structured, multi-page content as a single piece of content			•	•				
Web Content Management	WYSIWYG content editor	Support for WYSIWYG-based content editing				•	•			
Web Content Management	Multi-language content publication	Management of the publication of content in multiple languages		•	•			•		
Web Content Management	Text/rich-text editing	Support text and rich-text editing of content		•			•			
Web Content Management	File attachment support	Support the attachment of files to web-published content		•				•		
Web Content Management	File upload	Import of files for web publication via a web interface	•		•				•	
Web Content Management	Content tags	The association of identifying labels with web content		•					•	

Web Content Management	Permalinks	The creation of static links that reference web content		•						
Web Content Management	Content categorization	The management and association of content classifications	•	•				•		
Web Content Management	Preview content for web publication	Staging of web content prior to publication	•				•			
Web Content Management	Time-frame based publication (by date)	The publication of web content based on a specific time frame of availability						•	•	
Web Content Management	Community voting (polls)	The management and implementation of informal surveys			•					
Web Content Management	Content caching	The temporary caching of web content for efficient delivery			•					
Web Content Management	Syndication aggregation	Importing and merging of multiple syndication sources (for example, RSS reader)			•	•				
Web Content Management	Content commenting	Support for forum-based activity around pieces of content	•		•					
Collaboration	Forum support	Support for forum-based activities			•	•				
Collaboration	Email communication between users	Support for email between users of a content management system					•		•	
Weblog	Blogs	Support the publication of weblogs		•	•					
Weblog	Blog Entry Drafts	Support the management of blog drafts prior to publication		•						
Weblog	Weblog pings	Support for the informal weblog ping system		•	•					
Weblog	Weblog trackback	Support for the informal weblog		•						

	support	trackback system								
Weblog	Referrer tracking	Support for the informal weblog referrer tracking system		•						
Weblog	Blog entry moderation	Support for the preview of weblogs prior to publication		•						
Weblog	Author invite by e-mail	Support for the invitation of users via email		•						
Weblog	Word blacklist filter	The filtering of document content based on pre-specified lists of words		•						
Web Content Management	Bookmarks	Publication of bookmark lists to the web		•						
Business Process Management	Script-based Customization	Support for script-based processing of documents	•							
Records Management	Full-text search	The identification of content based on text contained within a document	•							
Records Management	Disposition Strategies	The management of document destruction based on a specified criteria	•							•
Records Management	Metadata Extraction	The identification of metadata within a document	•							
Records Management	Auto-classification	The classification of a document based on metadata	•							•
Records Management	Rules-based customization	Record management activity based on user-specified rules	•							
Records Management	Reporting	The creation of reports of record management activity	•							•
Records Management	File plans	The specification of required metadata and record management activity	•							

		(disposition strategy, etc.) based on document categorization								
Records Management	Auditing	The tracking of record management activities	•							
Records Management	Archival	The exporting of records	•							
Records Management	Lifecycle management	The management of a specified lifecycle stages for a given record	•							
Image Management	Metadata Extraction	The identification of metadata in images	•							
Image Management	Metadata Editing	The editing of the metadata associated with an image	•							
Image Management	Format conversion	The conversion of images between formats	•							
Image Management	Web service API	The access to image data via a web server API	•							
Image Management	E-mail alerts	Event notification based on e-mail	•							
Image Management	RSS alerts	Event notification based on RSS feeds	•							
Image Management	Integrated workflow	The routing of an image between users for management and approval	•							
Image Management	BPM rules integration	The processing of an image based on a pre-defined process	•							
Image Management	Metadata Search	The identification of images based on their metadata	•							
Image Management	Image album management	Using the album metaphor for organizing images	•				•			
Search	Open Search	Integration with the open search standard	•							
Search	MS Windows Desktop Search	Integration with the Microsoft Windows Desktop Search								•

Search	Full-text search	The identification of content based on text contained within a document	•	•	•		•		•	•
Document Management	Document conversion/transformation plug-ins	Support for conversion between document formats, including customization	•							
Document Management	Metadata Extraction	Support for the identification of metadata within a document							•	
Document Management	Metadata Editor	Support for the editing of the metadata associated with a document							•	
Document Management	Taxonomy organization	The specification and association of a taxonomy with documents			•					
Document Management	Content modeling/classification	The identification of specific content types and associated metadata	•	•					•	•
Document Management	Document spaces organization	The organization of content into distinct groupings	•							
Document Management	Document-level security	The specification of document security permissions on a per-document level	•						•	
Document Management	ACL-based permissions	The use of ACLs associated with specific documents to identify security permissions					•			
Document Management	Single Sign-on	The support of federated identity for user credentials	•							
Document Management	User, group, and role-based security	Support of general user, group, and role-based security	•							
Document Management	WebDAV	Support of the WebDAV specification for document editing	•				•			
Document	FTP upload	Support for FTP	•						•	

Management		transfer of files into the management system								
Document Managmeent	CIFS	Support for CIFS transfer of files into the management system	•							
Document Management	Print to folder (via print driver)	Support for a print from desktop to management system functionality							•	
Document Management	Scan to folder	Support for scan to management system functionality							•	
Document Management	Fax to folder	Support for Fax to management system functionality							•	
Document Management	Email to folder	Support for email to management system functionality							•	
Document Management	Bulk import API	Web service API for bulk upload of documents							•	
Document Management	File server import	Importing of documents from file system to management system								•
Document Management	PST/Outlook Import	Importing of documents from PST and MS Outlook to management system								•
Document Management	MS SharePoint portal import	Importing of documents from MS SharePoint portal to management system								•
Document Management	Enterprise content management system import	Importing of documents from ECM to management system								•
Document Management	Database import	Importing of documents from relational database management system to management system								•
Document Management	Instant messaging system import integration	Importing of instant messaging system to management system								•

Document Management	Extensibility for compliance and legal-holds	Support for compliance and legal-hold plug-ins								•
Document Management	Rules on folder (move based on metadata, trigger workflow, assign metadata, trigger events)	Rules-based actions associated with a folder							•	
Document Management	OCR (optical character recognition)	Identification of text in image formats							•	
Document Management	Document routing and approval	The movement of documents between users for processing and approval							•	
Document Management	Lifecycle management	The association of specified document stages with a given document	•							
Document Management	Check-in/out	Locking a document during editing	•						•	
Document Management	Document cross-linking	The validation of references between documents with the management system	•							
Document Management	Auditing	The tracking of document management activities	•						•	
Document Management	Versioning	The tracking of multiple versions of a particular document	•						•	
Document Management	Folder-based document organization	The use of a hierarchical structure of document groups for document organization	•				•	•	•	
Document Management	Digitally sign documents	The use of digital signatures to verify document authenticity							•	
Document	E-mail	The ability to send							•	

Management	document references (links)	references to documents on the management system								
Document Management	Document annotations and mark-up	The ability to add annotations and mark-ups to a document as metadata							•	
Content Repository	User, group, and/or role-based security	Document security based on user, group, and roles	•							
Content Repository	Document-level security	The application of permissions on a per-document level	•							
Content Repository	Document locking	The ability to temporarily reduce the permissions on a document during user editing	•							
Content Repository	document check-in/out	The ability to lock a document during user editing	•							
Content Repository	Lifecycle management	The association of a series of document stages and identification of a current stage to a given document	•							
Content Repository	Content streaming	The ability to portions of a document as needed	•							
Content Repository	document versioning	The management of multiple versions of a given document	•						•	
Content Repository	Script-based Customization	The customization of repository functionality using a scripting environment	•							
Content Repository	Workflow integration	Support for the movement of documents between users for processing and approval	•							
Content Repository	Metadata search	The identification of documents based on	•							

		document metadata								
Content Repository	Saved search queries	The management and re-use of specific search queries	•							
Content Repository	Search-oriented auditing	The tracking of the use of metadata search	•							
Content Repository	CIFS	The use of the CIFS interface for document import and export	•							
Content Repository	WebDAV	The use of the WebDAV interface for document import and export	•							
Content Repository	FTP	The use of the FTP interface for document import and export	•							
Content Repository	Web service API	The use of a web service API for access to content repository functionality	•							
Content Repository	Format conversion	The conversion of documents from one format to another	•						•	
Content Repository	Syndication	The use of syndication formats for sharing data with other systems	•							
Workflow Management	Event notification	The generation of notifications when specific events occur							•	
Workflow Management	Comment moderation	The ability to preview comments before publication	•	•						
Workflow Management	RSS event notification	Provide notification of certain events via the RSS format	•	•						
Core Functionality	System events	The ability for system components to generate and view events			•					
Core Functionality	Federated identity	The use of federated identity systems for			•					

		user credential management								
Core Functionality	Directory integration	The use of directory services for user credential management			•					
Core Functionality	User, group, role management	The use of user, group, and role management for document security	•	•	•	•	•	•	•	•
Core Functionality	User activity reports	The ability to generate reports to view audit records			•				•	
Core Functionality	Module extensibility	The ability to extend the functionality of a system using modules			•					
Core Functionality	Backup (Export)	The ability to export data from the system				•				
Course Management	Activity management	The management and publication of classroom activities				•				
Course Management	Calendar	The management and publication of calendar events				•				
Course Management	Journal management	The management and editing of journal entries				•				
Course Management	News publication	The management and publication of news items				•				
Course Management	Quiz support	The management and dispensing of on-line quizzes				•				
Course Management	Peer assessment	The management of peer assessment surveys				•				
Course Management	Survey support	The management and dispensing of surveys				•				
Course Management	Course Management	The management and publication of course materials				•				
Course Management	Chat support	Support for the interactive discussion between				•				

		users								
Course Management	Lesson management	The management and publication of lesson material				•				
Course Management	Assignment management	The management of lesson fulfillment by students				•				
Course Management	Dialogue support (2-person chat)	Chat support limited to private 2-person discussions				•				
Course Management	Wiki support	Support for wiki publication and management				•				
Course Management	Glossary publication (Definition lists)	Support for the management and publication of glossaries				•				

Appendix B
Source Code

ApiCmsImpl.java

```
package api;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.List;
import java.util.Map;
import java.util.Set;

import javax.persistence.EntityManager;

import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.queryParser.ParseException;
import org.apache.lucene.queryParser.QueryParser;
import org.apache.lucene.queryParser.QueryParser.Operator;
import org.apache.lucene.search.Query;
import org.hibernate.search.jpa.FullTextEntityManager;
import org.hibernate.search.jpa.FullTextQuery;
import org.hibernate.search.jpa.Search;

import processdocuments.ConvertPDFToPNG;
import processdocuments.ConvertToPDF;
import processdocuments.CountPDFPages;
import processdocuments.ExtractText;
import processdocuments.ResizeImage;

import document.Attribute;
import document.Document;
import document.Type;
import entity.Content;
import entity.ContentDAO;
import entity.ContentQuery;
import entity.EMFactory;

public class ApiCmsImpl {

	public static File convertDocToPDF(InputStream docFile, String suffix) 
throws IOException {
		File pdfRenditionFile = ConvertToPDF.convertToPDF(docFile, 
suffix);
		return pdfRenditionFile;
	}
	public static File convertDocxToPDF(InputStream docxFile, String suffix)
throws IOException {
		return convertDocToPDF(docxFile, suffix);
	}

	public static File convertPDFPageToPNG(File pdfFile, int pageNum) throws
FileNotFoundException, IOException {
		return ConvertPDFToPNG.convertPDFToPNG(new
FileInputStream(pdfFile),pageNum);
	}

	public static String convertPDFToText(File pdfFile) throws
FileNotFoundException, IOException {
		return ExtractText.extractText(new
FileInputStream(pdfFile),"pdf");
	}

	public static Content getContent(String key) {
		Content contentObj = ContentQuery.singleDoQuery("key:"+key);

		return contentObj;
	}

	public static String getLifecycle(String key) {
		String lifecycleState = getContent(key).getMetadata().get("life-
cycle-state");
		return lifecycleState;
	}

	public static List<Content> getLifecycleDocuments(String lifecycle) {
		String actualQuery = "life-cycle-state:"+lifecycle;
		List<Content> resultList = ContentQuery.doQuery(actualQuery);
		return resultList;
	}

	public static List<Content> getSubTypeChildren(String parentKey) {
		List<Content> childContentList =
ContentQuery.doQuery(Type.PARENT_KEY+":"+parentKey);
		return childContentList;
	}

	public static String[] getTypeHierarchy(String key) {
		String[] results = new
Document(getContent(key)).getTypeHierarchyKeyList();
		return results;
	}

	public static void index(String key, Map<String, String> metadata) {
		// currently integrated into setContent method

	}

	public static int numPagesInPDF(File pdfFile) throws IOException {
		return CountPDFPages.numPagesInPDF(pdfFile);
	}

	public static File resizeImage(File imageFile, int height, int width)
throws FileNotFoundException, IOException {
		return ResizeImage.resizePNG(new FileInputStream(imageFile),700,-
1);
	}
```

```
        public static File resizeImageByHeight(File imageFile, int height) throws
FileNotFoundException, IOException {
                return resizeImage(imageFile,height,-1);
        }

        public static File resizeImageByWidth(File imageFile, int width) throws
FileNotFoundException, IOException {
                return resizeImage(imageFile,-1,width);
        }

        public static List<Content> search(String queryString) {
                EntityManager em = EMFactory.getEntityManger();
                FullTextEntityManager fullTextEntityManager =
                        Search.getFullTextEntityManager(em);
                Query query;
                try {
                        QueryParser queryParser = new QueryParser("all-text-and-
attribute-values",new StandardAnalyzer());
                        queryParser.setDefaultOperator(Operator.AND);
                        query = queryParser.parse(queryString);
                } catch (ParseException e) {
                        e.printStackTrace();
                        throw new RuntimeException("ParseException parsing:
"+queryString);
                }
                FullTextQuery hibQuery =
fullTextEntityManager.createFullTextQuery(query,Content.class);

                @SuppressWarnings("unchecked")
                List<Content> results = (List<Content>)hibQuery.getResultList();
                return results;
        }

        public static Content setContent(String key, Map<String, String>
metadata,
                        InputStream content, int contentLength) {
                ContentDAO dao = new ContentDAO(EMFactory.getEntityManger(),true);
                Content resultContent = dao.create(key, metadata, content,
contentLength);
                return resultContent;
        }

        public static Content setContent(String key, Map<String, String>
metadata) {
                ContentDAO dao = new ContentDAO(EMFactory.getEntityManger(),true);
                Content resultContent = dao.create(key, metadata);
                return resultContent;
        }

        public static void setLifecycle(String key, String lifecycle) {
                Content contentObj = getContent(key);
                contentObj.getMetadata().put("life-cycle-state",lifecycle);
        }

}
```

Attribute.java

```
package document;

public class Attribute {

	private String uuid;
	private String name;
	private String value;
	private String typeKey;

	public Attribute(String uuid, String name, String value, String typeKey)
{
		this.uuid = uuid;
		this.name = name;
		this.value = value;
		this.typeKey = typeKey;
	}

	public String getUuid() {
		return uuid;
	}
	public void setUuid(String uuid) {
		this.uuid = uuid;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getValue() {
		return value;
	}
	public void setValue(String value) {
		this.value = value;
	}
	public String getTypeKey() {
		return typeKey;
	}
	public void setTypeKey(String typeKey) {
		this.typeKey = typeKey;
	}

}
```

Content.java

```
package entity;

import java.io.InputStream;
import java.sql.Blob;
import java.sql.SQLException;
import java.util.Date;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import java.util.Set;

import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.MapKey;
import javax.persistence.Transient;

import org.hibernate.annotations.CollectionOfElements;
import org.hibernate.lob.BlobImpl;
import org.hibernate.search.annotations.ClassBridge;
import org.hibernate.search.annotations.DocumentId;
import org.hibernate.search.annotations.Field;
import org.hibernate.search.annotations.FieldBridge;
import org.hibernate.search.annotations.Index;
import org.hibernate.search.annotations.Indexed;

@Entity
@Indexed
public class Content {

       private Long id;
       private String key;
       private Map<String,String> metadata;
    private Blob content;

       @Id @GeneratedValue
       @DocumentId
       public Long getId() {
              return id;
       }

       @SuppressWarnings("unused")
       private void setId(Long id) {
              this.id = id;
       }

       @Column(name="`key`",length=512)
       @org.hibernate.annotations.Index(name = "key_IDX")
       @Field(index=Index.TOKENIZED)
       public String getKey() {
              return key;
       }

       public void setKey(String key) {
              this.key = key;
       }

       public String[] findKeysWithPrefix(String prefix) {
              if(this.metadata==null) {
                     return new String[0];
              }

              Set<String> resultKeySet = new HashSet<String>();

              Set<String> keySet = this.metadata.keySet();
              for(String key: keySet) {
                     if(key.startsWith(prefix)) {
                            resultKeySet.add(key);
                     }
```

```
		}
		String[] resultKeyArray = (String[])resultKeySet.toArray();
		return resultKeyArray;
	}

	@CollectionOfElements
	@Column(length=Integer.MAX_VALUE-1)
	@Field()
	@FieldBridge(impl=MetadataFieldBridge.class)
	public synchronized Map<String, String> getMetadata() {
		if(metadata==null) {
			metadata = new HashMap<String,String>();
		}
		return metadata;
	}

	public void setMetadata(Map<String, String> metadata) {
		this.metadata = metadata;
	}

	@Lob // should default to lazy load
	@Column(length=Integer.MAX_VALUE - 1,name="content")
	private Blob getContentBlob() {
		return content;
	}
	private void setContentBlob(Blob content) {
		this.content = content;
	}
	@Transient
	public InputStream getContent() throws SQLException {
		return getContentBlob().getBinaryStream();
	}
	public void setContent(InputStream contentIS, int contentLength) {
		Blob blob = new BlobImpl(contentIS, contentLength);
		setContentBlob(blob);
	}

}
```

ContentDAO.java

```
package entity;

import java.io.InputStream;
import java.util.Date;
import java.util.List;
import java.util.Map;

import javax.persistence.EntityManager;
import javax.persistence.EntityNotFoundException;
import javax.persistence.EntityTransaction;
import javax.transaction.Transaction;

public class ContentDAO {

	private EntityManager em;
	private EntityTransaction tx;
	private boolean manageTransactions;
	public ContentDAO(EntityManager em, boolean manageTransactions) {
		init(em,manageTransactions);
	}

	public ContentDAO(EntityManager em) {
		init(em,true);
	}

	private void init(EntityManager em, boolean manageTransactions) {
		this.manageTransactions = manageTransactions;
		this.em = em;
	}

	private void joinTransaction() {
		if(manageTransactions==true) {
			EntityTransaction tx = em.getTransaction();
			if(tx.isActive()==false) {
				tx.begin();
			}
		}
	}

	private void commitTransaction() {
		if(manageTransactions==true) {
		if(tx==null) { // I think we're already in trouble here
			tx = em.getTransaction();
		} else {
			if(tx.isActive()) {
				tx.commit();
			}
		}
		}
	}

	public Content updateMetadata(String key, Map<String,String> metadata)
throws EntityNotFoundException {
		joinTransaction();
		Content result = _updateMetadata(key,metadata);
		commitTransaction();
		return result;
	}

	public int delete(String key) throws EntityNotFoundException {
		joinTransaction();
		int result = _delete(key);
		commitTransaction();
		return result;
	}

	public Content create(String key, Map<String,String> metadata) {
		return create(key, metadata, null, 0); // just need null input
stream to work (length is ignored)
	}

	public Content create(String key, Map<String, String> metadata,
InputStream contentIS, int contentLength) throws EntityNotFoundException {
		joinTransaction();
		Content result = _create(key, metadata, contentIS, contentLength);
		commitTransaction();
		return result;
	}
	public Content read(String key) throws EntityNotFoundException {
		//joinTransaction(); // no transaction required to read data
		Content result = _read(key);
		//commitTransaction();
		return result;
	}

	private Content _create(String key, Map<String,String> metadata,
InputStream contentIS, int contentLength) {
		try {
			_delete(key);
		} catch(EntityNotFoundException ex) {
			// do nothing; just making sure its gone before creating
new one
		}
		Content to = new Content();
		to.setKey(key);
		to.setMetadata(metadata);
		if(contentIS!=null) {
			to.setContent(contentIS, contentLength);
		}

		em.persist(to);
		return to;
	}

	private Content _read(String key) {
		String query = "from Content to " +
			"where to.key=:key ";
		@SuppressWarnings("unchecked")
		List<Content> results = em.createQuery(query)
			.setParameter("key", key)
			.getResultList();

		Content resultObj = null;
		if(results.size()>0) {
			resultObj = results.get(0);
		} else {
			throw new EntityNotFoundException("entity with key
\""+key+"\" not found");
		}
		return resultObj;
	}

	private Content _updateMetadata(String key, Map<String,String> metadata)
{
		Content resultObj = _read(key);
		resultObj.setMetadata(metadata);
		em.persist(resultObj);

		return resultObj;

	}

	private int _delete(String key) {
		String query = "from Content to "
			+ "where key=:key "
			;
		@SuppressWarnings("unchecked")
		List<Content> results = em.createQuery(query)
			.setParameter("key", key)
			.getResultList();
		for(Content result: results) {
			em.remove(result);
		}
		return results.size();
	}

	public List<Content> list(String prefix, String marker, int maxKeys) {
		// may not participate in a transaction
		String query = "from Content to "
				+ "where to.key like :prefix "
				+ "and to.key > :marker "
				+ "order by to.key asc"
			;
		@SuppressWarnings("unchecked")
		List<Content> results = em.createQuery(query)
			.setParameter("prefix", prefix+"%")
			.setParameter("marker", marker)
			.setMaxResults(maxKeys)
			.getResultList();
		return results;
	}
}
```

ContentQuery.java

```
package entity;

import java.util.List;
import javax.persistence.EntityManager;

import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.queryParser.ParseException;
import org.apache.lucene.queryParser.QueryParser;
import org.apache.lucene.queryParser.QueryParser.Operator;
import org.apache.lucene.search.Query;
import org.hibernate.search.jpa.FullTextEntityManager;
import org.hibernate.search.jpa.FullTextQuery;
import org.hibernate.search.jpa.Search;

import api.ApiCmsImpl;

public class ContentQuery {
	static EntityManager em = EMFactory.getEntityManger();

	public static List<Content> doQuery(String queryString) {

		return ApiCmsImpl.search(queryString);
	}

	public static Content singleDoQuery(String queryString) {
		List<Content> contentList = doQuery(queryString);
		if(contentList.size()>0) {
			return contentList.get(0);
		} else {
			return null;
		}
	}

}
```

ConvertPDFToPNG.java

```
package processdocuments;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;


//C:\temp>"c:\Program Files\gs\gs8.64\bin\gswin32c.exe" -dSAFER -dBATCH -
dNOPAUSE
//-sDEVICE=png16m -dGraphicsAlphaBits=4 -r300 -dFirstPage=99999 -dLastPage=9999
-s
//OutputFile=tiger-%05d.png test.pdf


public class ConvertPDFToPNG {

       final static String CMD_ARGS[] = {
              "c:\\Program Files\\gs\\gs8.64\\bin\\gswin32c.exe", "-dSAFER", "-
dBATCH",
              "-dNOPAUSE", "-sDEVICE=png16m", "-dGraphicsAlphaBits=4", "-r300",
              "-dFirstPage=%PAGE%", "-dLastPage=%PAGE%", "-
sOutputFile=%RESULT%", "%INPUT%"
       };


       /**
        * @param args
        */
       public static void main(String[] args) throws Exception {

              String filepath = "C:/temp/test.pdf";
              convertPDFToPNG(new FileInputStream(filepath), 13);

       }

       public static InputStream convertPDFToPNG(InputStream pdfStream, int
pageNum) throws IOException {
              File inputTmpFile = File.createTempFile("convert_pdf_to_png-",
".pdf");
              File resultTmpFile = File.createTempFile("convert_pdf_to_png-",
".png");

              OutputStream inputOS = new FileOutputStream(inputTmpFile);

              byte[] buffer = new byte[1024];
              int bytesRead = 0;
              bytesRead = pdfStream.read(buffer);
              while(bytesRead>0) {
                     inputOS.write(buffer);
                     bytesRead = pdfStream.read(buffer);
              }
              inputOS.close();

              String[] cmd = new String[11];
              System.arraycopy(CMD_ARGS, 0, cmd, 0, 11);

              cmd[7] = cmd[7].replace("%PAGE%", Integer.toString(pageNum));
```

```
		cmd[8] = cmd[8].replace("%PAGE%", Integer.toString(pageNum));
		cmd[9] = cmd[9].replace("%RESULT%", resultTmpFile.getPath());
		cmd[10] = cmd[10].replace("%INPUT%", inputTmpFile.getPath());

		Runtime runtime = Runtime.getRuntime();
		Process process = runtime.exec(cmd);
		try {
			process.waitFor();
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
//		InputStream is = process.getInputStream();

		return new FileInputStream(resultTmpFile.getPath());

	}

}
```

ConvertToPDF.java

```
package processdocuments;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

// C:\temp>"c:\Program Files\gs\gs8.64\bin\gswin32c.exe" -dSAFER -dBATCH -
dNOPAUSE
// -sDEVICE=png16m -dGraphicsAlphaBits=4 -r300 -dFirstPage=99999 -
dLastPage=9999 -s
// OutputFile=tiger-%05d.png test.pdf

public class ConvertToPDF {

	static final String CMD_ARGS[] = {"c:\\Program Files\\OpenOffice.org
3\\program\\swriter.exe", "-invisible"
		, "macro:///Standard.Module1.ConvertWordToPDF(%%)"};

	/**
	 * @param args
	 */
	public static void main(String[] args) throws Exception {
		// TODO Auto-generated method stub

		String filepath = "C:\\temp\\darknet5.doc.doc";

		InputStream resultInputStream = convertToPDF(new
FileInputStream(filepath), ".doc");
	}

	public static InputStream convertToPDF(InputStream fileIS, String suffix)
throws IOException { // pageNum is 1-indexed
```

```
        File inputTmpFile = File.createTempFile("convert2pdf-input_file-",
suffix);

        String resultTmpFilename = inputTmpFile.getPath();
        resultTmpFilename = resultTmpFilename.replace(suffix, ".pdf");
        File resultTmpFile = new File(resultTmpFilename);

        byte[] buffer = new byte[1024];

        // write input file to tmp file
        OutputStream inputOS = new FileOutputStream(inputTmpFile);
        int bytesRead = 0;
        bytesRead = fileIS.read(buffer);
        while(bytesRead>0) {
            inputOS.write(buffer);
            bytesRead = fileIS.read(buffer);
        }

        System.out.println(inputTmpFile);

        String[] cmd = new String[3];
        System.arraycopy(CMD_ARGS, 0, cmd, 0, 3);
        cmd[2] = cmd[2].replace("%%", inputTmpFile.getPath());

        Runtime runtime = Runtime.getRuntime();
        Process process = runtime.exec(cmd);
        try {
            process.waitFor();
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        System.out.println(resultTmpFile);


        return new FileInputStream(resultTmpFile);
    }

}
```

CountPDFPages.java

```
package processdocuments;
import java.io.*;
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class CountPDFPages {

    static final String cmdArgs[] = { "c:\\Program
Files\\gs\\gs8.64\\bin\\gswin32c.exe",
            "-dSAFER", "-dBATCH", "-dNOPAUSE", "-
dFirstPage=99999999999999999999"};

    public static void main(String args[]) throws IOException {
```

```
		String pathname = "c:\\temp\\test.pdf";

		int numPagesInPDF = numPagesInPDF(new File(pathname));

		System.out.println(numPagesInPDF);

	}

	public static int numPagesInPDF(File file) throws IOException {

		if(file.isFile()==false) {
			throw new FileNotFoundException("The file does not exist: 
"+file.getPath());
		}

		String cmd[] = new String[6];
		System.arraycopy(cmdArgs, 0, cmd, 0, 5);
		cmd[5] = file.getPath();

		Runtime runtime = Runtime.getRuntime();
		Process process = runtime.exec(cmd);
		InputStream is = process.getInputStream();

		InputStreamReader isr = new InputStreamReader(is);
		BufferedReader br = new BufferedReader(isr);
		String line;

		//		System.out.printf("Output of running %s is:",
		//				Arrays.toString(args));

		int numPagesInPDF = 0;

		String regex
		= "^Requested FirstPage is greater than the number of pages in the 
file: (\\d+)$";
		Pattern p = Pattern.compile(regex);
		while ((line = br.readLine()) != null) {
			Matcher m = p.matcher(line);
			if(m.matches()) {
				numPagesInPDF = Integer.parseInt(m.group(1));
				break;
			}
		}

		return numPagesInPDF;
	}
}
```

DateUtils.java

```
package util;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class DateUtils {
	final static String formatPattern = "yyyyMMddHHmmssSSSZ";
	static SimpleDateFormat formatter = new SimpleDateFormat(formatPattern);

	public static void main(String args[]) {
		String dateString = convertDateToString(new Date());
		System.out.println(dateString);

		Date resultDate = convertStringToDate(dateString);
		System.out.println(resultDate);

	}

	public static String convertDateToString(Date date) {
		if(date==null) return null;
		return formatter.format(date);
	}

	public static Date convertStringToDate(String dateString) {
		if(dateString==null) return null;
		try {
			return formatter.parse(dateString);
		} catch (ParseException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
			throw new RuntimeException("date string is not formatted
correctly");
		}
	}
}
```

Document.java

```
package document;

import java.io.InputStream;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Map;
import java.util.Set;
import java.util.UUID;

import util.DateUtils;
import util.StringUtils;

import entity.Content;

public class Document {

	private String key;
	final public static String KEY = "key";
	private String lifeCycleState;
	final public static String LIFE_CYCLE_STATE = "life-cycle-state";
	private Date captureDate;
	final public static String CAPTURE_DATE = "capture-date";
```

```
        private Date processingCompleteDate;
        final public static String PROCESSING_COMPLETE_DATE = "processing-
capture-date";
        private Date approvalDate;
        final public static String APPROVAL_DATE = "approval-date";
        private String approvingUserUUID;
        final public static String APPROVING_USER_UUID = "approving-user-uuid";
        private String originalRenditionKey;
        final public static String ORIGINAL_RENDITION_KEY = "original-rendition-
key";
        private String pdfRenditionKey;
        final public static String PDF_RENDITION_KEY = "pdf-rendition-key";
        private String textContent;
        final public static String TEXT_CONTENT = "text-content";

        private String submitterUserUUID;
        final public static String SUBMITTER_USER_UUID = "submitter-user-uuid";

        private String[] typeHierarchyKeyList;
        final public static String TYPE_HIERARCHY_KEY_LIST = "type-hierarchy-key-
list";
        private String[] typeHierarchyNameList;
        final public static String TYPE_HIERARCHY_NAME_LIST = "type-hierarchy-
name-list";

        private String[] imageRenditionKeyList;
        final public static String IMAGE_RENDITION_KEY_LIST = "image-renditon-
key-list";
        private String thumbnailImage;
        final public static String THUMBNAIL_IMAGE = "thumbnail-image";

        final public static String ATTRIBUTE_UUID_LIST = "attribute-uuid-list";
        private Attribute[] attributeList;
        final public static String ATTRIBUTE_NAME_PREFIX = "attribute-name-";
        final public static String ATTRIBUTE_VALUE_PREFIX = "attribute-value-";
        final public static String ATTRIBUTE_TYPE_KEY_PREFIX = "attribute-type-
key-";


        private Content documentContent; // reference to content object
        private Map<String,String> metadata;

        public Document() { }

        public Document(Content documentContent) {
                this.documentContent = documentContent; // save reference to get
content stream when needed
                this.metadata = documentContent.getMetadata();

                this.key = documentContent.getKey();

                this.lifeCycleState = metadata.get(LIFE_CYCLE_STATE);
                this.captureDate =
DateUtils.convertStringToDate(metadata.get(CAPTURE_DATE));
                this.processingCompleteDate =
DateUtils.convertStringToDate(metadata.get(PROCESSING_COMPLETE_DATE));
                this.approvalDate =
DateUtils.convertStringToDate(metadata.get(APPROVAL_DATE));
                this.approvingUserUUID = metadata.get(APPROVING_USER_UUID);
                this.originalRenditionKey = metadata.get(ORIGINAL_RENDITION_KEY);
                this.pdfRenditionKey = metadata.get(PDF_RENDITION_KEY);
                this.textContent = metadata.get(TEXT_CONTENT);
                this.submitterUserUUID = metadata.get(SUBMITTER_USER_UUID);
                typeHierarchyKeyList =
metadata.get(TYPE_HIERARCHY_KEY_LIST).split("\t");
                typeHierarchyNameList =
metadata.get(TYPE_HIERARCHY_NAME_LIST).split("\t");

                this.imageRenditionKeyList =
metadata.get(IMAGE_RENDITION_KEY_LIST).split("\t");
                this.thumbnailImage = metadata.get(THUMBNAIL_IMAGE);
                this.attributeList = parseAttributeMetadata(documentContent);
        }

        public Content getContentObject() {

                Content documentContent = new Content();
                Map<String,String>  metadata = new HashMap<String,String>();
                documentContent.setMetadata(metadata);

                documentContent.setKey(key);

                metadata.put(LIFE_CYCLE_STATE,this.lifeCycleState);
                metadata.put(CAPTURE_DATE,
DateUtils.convertDateToString(this.captureDate));
                metadata.put(PROCESSING_COMPLETE_DATE,
DateUtils.convertDateToString(this.processingCompleteDate));

        metadata.put(APPROVAL_DATE,DateUtils.convertDateToString(this.approvalDat
e));
                metadata.put(APPROVING_USER_UUID, this.approvingUserUUID);
                metadata.put(ORIGINAL_RENDITION_KEY, this.originalRenditionKey);
                metadata.put(PDF_RENDITION_KEY, this.pdfRenditionKey);
                metadata.put(TEXT_CONTENT, this.textContent);
                metadata.put(SUBMITTER_USER_UUID, this.submitterUserUUID);

                metadata.put(TYPE_HIERARCHY_KEY_LIST,
StringUtils.join(typeHierarchyKeyList, "\t"));
                metadata.put(TYPE_HIERARCHY_NAME_LIST,
StringUtils.join(typeHierarchyNameList, "\t"));

                metadata.put(IMAGE_RENDITION_KEY_LIST,
StringUtils.join(this.imageRenditionKeyList,"\t"));
                metadata.put(THUMBNAIL_IMAGE, this.thumbnailImage);
                putAttributeMetadata(metadata, attributeList);

                return documentContent;
        }

        private Attribute[] parseAttributeMetadata(Content documentContent) {
                Map<String,String> metadata = documentContent.getMetadata();
                String attributeUuidListString =
metadata.get(ATTRIBUTE_UUID_LIST);
                String[] attributeUuidList = attributeUuidListString.split("\t");
                List<Attribute> attributeList = new ArrayList<Attribute>();
                for(String attributeUuid: attributeUuidList) {
                        String nameKey = ATTRIBUTE_NAME_PREFIX+attributeUuid;
                        String valueKey = ATTRIBUTE_VALUE_PREFIX+attributeUuid;
                        String typeKeyKey =
ATTRIBUTE_TYPE_KEY_PREFIX+attributeUuid;
                        String name = metadata.get(nameKey);
                        String value = metadata.get(valueKey);
                        String typeKey = metadata.get(typeKeyKey);
                        Attribute curAttribute = new
Attribute(attributeUuid,name,value,typeKey);
```

```
            attributeList.add(curAttribute);
        }
        return attributeList.toArray(new Attribute[attributeList.size()]);
    }

    private void putAttributeMetadata(Map<String,String> metadata,
Attribute[] attributeList) {
        StringBuilder uuidSB = new StringBuilder();
        boolean first = true;
        for(Attribute curAttribute: attributeList) {
            String attributeUuid = curAttribute.getUuid();
            if(attributeUuid==null) {
                attributeUuid = UUID.randomUUID().toString();
            }
            String name = curAttribute.getName();
            String value = curAttribute.getValue();
            String typeKey = curAttribute.getTypeKey();
            String nameKey = ATTRIBUTE_NAME_PREFIX+attributeUuid;
            String valueKey = ATTRIBUTE_VALUE_PREFIX+attributeUuid;
            String typeKeyKey =
ATTRIBUTE_TYPE_KEY_PREFIX+attributeUuid;
            metadata.put(nameKey,name);
            metadata.put(valueKey,value);
            metadata.put(typeKeyKey,typeKey);
            if(first==false) {
                uuidSB.append("\t");
            } else {
                first=false;
            }
            uuidSB.append(attributeUuid);
        }

        String attributeUuidList = uuidSB.toString();
        metadata.put(ATTRIBUTE_UUID_LIST, attributeUuidList);
    }

    public String getKey() {
        return key;
    }

    public void setKey(String key) {
        this.key = key;
    }

    public String getLifeCycleState() {
        return lifeCycleState;
    }

    public void setLifeCycleState(String lifeCycleState) {
        this.lifeCycleState = lifeCycleState;
    }


    public Date getCaptureDate() {
        return captureDate;
    }

    public void setCaptureDate(Date captureDate) {
        this.captureDate = captureDate;
    }

    public Date getProcessingCompleteDate() {
        return processingCompleteDate;
```

```java
}

public void setProcessingCompleteDate(Date processingCompleteDate) {
	this.processingCompleteDate = processingCompleteDate;
}

public Date getApprovalDate() {
	return approvalDate;
}

public void setApprovalDate(Date approvalDate) {
	this.approvalDate = approvalDate;
}

public String getApprovingUserUUID() {
	return approvingUserUUID;
}

public void setApprovingUserUUID(String approvingUserUUID) {
	this.approvingUserUUID = approvingUserUUID;
}

public String getOriginalRenditionKey() {
	return originalRenditionKey;
}

public void setOriginalRenditionKey(String originalRenditionKey) {
	this.originalRenditionKey = originalRenditionKey;
}

public String getPdfRenditionKey() {
	return pdfRenditionKey;
}

public void setPdfRenditionKey(String pdfRenditionKey) {
	this.pdfRenditionKey = pdfRenditionKey;
}

public String getTextContent() {
	return textContent;
}

public void setTextContent(String textContent) {
	this.textContent = textContent;
}

public String[] getImageRenditionKeyList() {
	return imageRenditionKeyList;
}

public void setImageRenditionKeyList(String[] imageRenditionKeyList) {
	this.imageRenditionKeyList = imageRenditionKeyList;
}

public String getSubmitterUserUUID() {
	return submitterUserUUID;
}

public void setSubmitterUserUUID(String submitterUserUUID) {
	this.submitterUserUUID = submitterUserUUID;
}

public InputStream getContentStream() throws SQLException {
		if(documentContent!=null) {
			return this.documentContent.getContent();
		} else {
			return null;
		}
	}

	public String[] getTypeHierarchyKeyList() {
		return typeHierarchyKeyList;
	}

	public void setTypeHierarchyKeyList(String[] typeHierarchyKeyList) {
		this.typeHierarchyKeyList = typeHierarchyKeyList;
	}

	public String[] getTypeHierarchyNameList() {
		return typeHierarchyNameList;
	}

	public void setTypeHierarchyNameList(String[] typeHierarchyNameList) {
		this.typeHierarchyNameList = typeHierarchyNameList;
	}

	public Attribute[] getAttributeList() {
		return attributeList;
	}

	public void setAttributeList(Attribute[] attributeList) {
		this.attributeList = attributeList;
	}

	public String getThumbnailImage() {
		return thumbnailImage;
	}

	public void setThumbnailImage(String thumbnailImage) {
		this.thumbnailImage = thumbnailImage;
	}

}
```

edit_type.jsp

```
<%

String editKey = request.getParameter("edit_key");
TypeAttribute[] curAttributes = null;
Type curType=null;
if(editKey!=null) {
	Content contentObj = ApiCmsImpl.getContent(editKey);
	curType = new Type(contentObj);
	curAttributes = curType.getTypeAttributeArray();
}

%>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import="entity.ContentQuery"%>
<%@page import="entity.Content"%>
<%@page import="document.Type"%>
<%@page import="document.TypeAttribute"%>
<%@page import="api.ApiCmsImpl"%><html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>
<form action="EditTypeServlet" method="post">
Type name: <input type="text" name="type_name" value="<%
if(curType!=null) {
	out.print(curType.getName());
}
%>"/> <br />
<% for(int i=0; i<15; i++) { %>
Name:
<input type="text" name="name_<%= i %>" value="<%
if(curAttributes!=null && curAttributes.length>i) {
	String attribName = curAttributes[i].getAttributeName();
	out.print(attribName);
}
%>" />
Type:
<select name="type_<%= i %>" >
<option value="text" <%
if(curAttributes!=null && curAttributes.length>i) {
	if(curAttributes[i].getAttributeType()!=null &&
curAttributes[i].getAttributeType().equals("text")) {
		out.print("SELECTED");
	}
}
%>>Text</option>
<option value="date" <%
if(curAttributes!=null && curAttributes.length>i) {
	if(curAttributes[i].getAttributeType()!=null &&
curAttributes[i].getAttributeType().equals("date")) {
		out.print("SELECTED");
	}
}
%>>Date</option>
<option value="choice" <%
if(curAttributes!=null && curAttributes.length>i) {
	if(curAttributes[i].getAttributeType()!=null &&
curAttributes[i].getAttributeType().equals("choice")) {
		out.print("SELECTED");
	}
}
%>>Choice</option>
</select>
Options:
<textarea name="options_<%= i %>"><%
if(curAttributes!=null && curAttributes.length>i) {
	String attribOptions = curAttributes[i].getAttributeOptions();
	out.print(attribOptions);
}
%></textarea>
<br/>
<% } %>

<br/>

<input type="hidden" name="parent_key"
value="<%=request.getParameter("parent_key") %>"/>
<input type="hidden" name="edit_key" value="<%=editKey %>"/>

<input type="submit" /> <a href="./list_types.jsp">Cancel</a>
</form>
</body>
</html>
```

EMFactory.java

```
package entity;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class EMFactory {


	static EntityManagerFactory emf =
Persistence.createEntityManagerFactory("sample");
//	EntityManager em = emf.createEntityManager(); // Retrieve an application
managed entity manager

	public static EntityManager getEntityManger() {
		return emf.createEntityManager();
	}

	public static void close() {
		emf.close();
	}
}
```

ExtractText.java

```
package processdocuments;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
```

```
//"c:\Program Files\Windows Platform SDK\bin\FiltDump.Exe" -b <file> > <out-
file>


public class ExtractText {

        final static String CMD_ARGS[] = {
                "C:\\WINDOWS\\system32\\cmd.exe",
                "/C",
                "\"c:\\Program Files\\Windows Platform SDK\\bin\\FiltDump.Exe\"",
                "-b",
                "%INPUT%",
                ">",
                "%RESULT%"
        };



        /**
         * @param args
         */
        public static void main(String[] args) throws Exception {

                String filepath = "C:/temp/parentinvguid.doc.doc";
                InputStream resultIS = extractText(new
FileInputStream(filepath),".doc");
                BufferedReader br = new BufferedReader(new
InputStreamReader(resultIS));
                while(br.ready()) {
                        String line = br.readLine();
                        System.out.println(line);
                }
        }

        public static InputStream extractText(InputStream fileIS,String
extension) throws IOException {
                File inputTmpFile = File.createTempFile("extract_text-",
extension);
                File resultTmpFile = File.createTempFile("extract_text-", ".txt");

                OutputStream inputOS = new FileOutputStream(inputTmpFile);

                System.out.println(inputTmpFile.getPath());

                byte[] buffer = new byte[1024];
                int bytesRead = 0;
                bytesRead = fileIS.read(buffer);
                while(bytesRead>0) {
                        inputOS.write(buffer);
                        bytesRead = fileIS.read(buffer);
                }
                inputOS.close();

                String[] cmd = new String[CMD_ARGS.length];
                System.arraycopy(CMD_ARGS, 0, cmd, 0, CMD_ARGS.length);

                //              cmd[3] = cmd[3].replace("%INPUT%",
inputTmpFile.getPath());
                //              cmd[5] = cmd[5].replace("%RESULT%",
resultTmpFile.getPath());
                cmd[4] = cmd[4].replace("%INPUT%", inputTmpFile.getPath());
                cmd[6] = cmd[6].replace("%RESULT%", resultTmpFile.getPath());
```

```
            for(String arg: cmd) {
                  System.out.print(arg+" ");
            }
            System.out.println();

            Runtime runtime = Runtime.getRuntime();
            Process process = runtime.exec(cmd);
            try {
                  process.waitFor();
            } catch (InterruptedException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }
            //            InputStream is = process.getInputStream();

            System.out.println(resultTmpFile.getPath());

            return new FileInputStream(resultTmpFile.getPath());

      }

}
```

FileUtils.java

```
package util;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;

public class FileUtils {
      public static File writeToFile(InputStream is) {
            try {
            File resultFile = File.createTempFile("tmpFile", ".tmp");
            OutputStream os = new FileOutputStream(resultFile);
            byte[] buffer = new byte[1024];
            int bytesRead;
            bytesRead = is.read(buffer);
            while(bytesRead>0) {
                  os.write(buffer);
                  bytesRead = is.read(buffer);
            }
            return resultFile;
            } catch(IOException ex) {
                  ex.printStackTrace();
                  throw new RuntimeException("problem creating stream-based 
file");
            }
      }
}
```

index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<META
http-equiv="refresh"
content="0;URL=links.html">
<title>Insert title here</title>
</head>
<body>

</body>
</html>
```

LifeCycle.java

```
package document;

public class LifeCycle {
       final public static String PENDING_CAPTURE = "pending-capture";
       final public static String PENDING_PROCESSING = "pending-processing";
       final public static String PROCESSING = "processing";
       final public static String PENDING_APPROVAL = "pending-approval";
       final public static String PUBLISHED = "published";
       final public static String ARCHIVED = "archived";

}
```

links.java

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="list_types.jsp">Edit Document Content Model</a><br/>
<a href="select_type.jsp">Add Document</a><br/>
<a href="process_docs.jsp">Process Documents</a><br/>
<a href="search_results.jsp?review=1">Review Documents</a><br/>
<a href="search.jsp">Search Documents</a><br/>

</body>
</html>
```

list_types.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="list_types.jsp">Edit Document Content Model</a><br/>
<a href="select_type.jsp">Add Document</a><br/>
<a href="process_docs.jsp">Process Documents</a><br/>
<a href="search_results.jsp?review=1">Review Documents</a><br/>
<a href="search.jsp">Search Documents</a><br/>

</body>
</html>
```

MetadataFieldBridge.java

```
package entity;

import java.util.Map;
import java.util.Set;
import java.util.Map.Entry;

import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;
import org.apache.lucene.document.Field.Index;
import org.apache.lucene.document.Field.Store;
import org.hibernate.search.bridge.FieldBridge;
import org.hibernate.search.bridge.LuceneOptions;

public class MetadataFieldBridge implements FieldBridge {

      public void set(String arg0, Object metadataObj, Document doc,
LuceneOptions opts) {
             @SuppressWarnings("unchecked")
             Map<String,String> metadata = (Map<String,String>)metadataObj;
             // ignore opts

             StringBuilder allMetadataKeys = new StringBuilder();
             StringBuilder allMetadataValues = new StringBuilder();
             StringBuilder allAttributeValues = new StringBuilder();

             Set<Entry<String,String>> entries = metadata.entrySet();
             for(Entry<String,String> entry: entries) {
                    String key = entry.getKey();
                    String value = entry.getValue();

                    if(key==null || value==null) {
                          continue;
                    }

                    Field field = new Field(key,value,Store.NO,Index.ANALYZED);
                    doc.add(field);

                    allMetadataKeys.append(key+"\t");
                    allMetadataValues.append(value+"\t");

        if(key.startsWith(document.Document.ATTRIBUTE_VALUE_PREFIX)) {
                            allAttributeValues.append(value);
                            allAttributeValues.append("\t");
                    }
            }
            Field amkField = new Field("all-metadata-
keys",allMetadataKeys.toString(),Store.NO,Index.ANALYZED);
            doc.add(amkField);
            Field amvField = new Field("all-metadata-
values",allMetadataValues.toString(),Store.NO,Index.ANALYZED);
            doc.add(amvField);
            if(allAttributeValues.length()>0) { // check if document object
                    String allAttributeValuesString =
allAttributeValues.toString();
                    Field aavField = new Field("all-attribute-
values",allAttributeValuesString,Store.NO,Index.ANALYZED);
                    doc.add(aavField);
                    String allTextAndAttributeValues =
metadata.get(document.Document.TEXT_CONTENT) + allAttributeValuesString;
                    Field allTextandAttributeValuesField = new Field("all-text-
and-attribute-values",allTextAndAttributeValues,Store.NO,Index.ANALYZED);
                    doc.add(allTextandAttributeValuesField);
            }
    }

}
```

process_docs.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import="entity.ContentQuery"%>
<%@page import="document.Document"%>
<%@page import="document.LifeCycle"%>
<%@page import="java.util.List"%>
<%@page import="entity.Content"%>
<%@page import="document.Rendition"%>
<%@page import="processdocuments.ConvertToPDF"%>
<%@page import="org.apache.tomcat.util.http.MimeMap"%>
<%@page import="java.io.InputStream"%>
<%@page import="java.util.UUID"%>
<%@page import="processdocuments.CountPDFPages"%>
<%@page import="java.io.File"%>
<%@page import="java.io.FileInputStream"%>
<%@page import="util.FileUtils"%>
<%@page import="javax.persistence.EntityManager"%>
<%@page import="entity.EMFactory"%>
<%@page import="javax.persistence.EntityTransaction"%>
<%@page import="entity.ContentDAO"%>
<%@page import="processdocuments.ConvertPDFToPNG"%>
<%@page import="java.util.ArrayList"%>
<%@page import="processdocuments.ResizeImage"%>
```

```
<%@page import="processdocuments.ExtractText"%>
<%@page import="api.ApiCmsImpl"%><html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>

</body>
</html>
<%

String pendingProcessingQuery =
Document.LIFE_CYCLE_STATE+":"+LifeCycle.PENDING_PROCESSING;
System.out.println("pending processing query: "+pendingProcessingQuery);
List<Content> resultList = ContentQuery.doQuery(pendingProcessingQuery);
EntityManager em = EMFactory.getEntityManger();
EntityTransaction tx = em.getTransaction();
tx.begin();
for(Content curContentObj: resultList) {
        Document curDocument = new Document(curContentObj);
        System.out.println(curDocument.getKey());
        System.out.println(curDocument.getOriginalRenditionKey());

        String query = Document.KEY+":"+curDocument.getOriginalRenditionKey();
        System.out.println(query);
        Content curRenditionContentObj = ContentQuery.singleDoQuery(query);
        if(curRenditionContentObj==null)
                continue;
        Rendition curRendition = new Rendition(curRenditionContentObj);
        String filename = curRendition.getFilename();
        System.out.println(filename);
        String contentType = curRendition.getContentType();
        System.out.println(contentType);
        int contentLength = curRendition.getContentLength();

        if(curRendition==null || contentType==null)
                continue;

        File pdfRenditionFile = null;

        boolean recognizedFile = false;
        if(contentType.equals("application/pdf")) {
                System.out.println("PDF!");
                curDocument.setPdfRenditionKey(curRendition.getKey());
                pdfRenditionFile =
FileUtils.writeToFile(curRendition.getContentIS());
                recognizedFile = true;
        } else if(contentType.equals("application/vnd.openxmlformats-
officedocument.wordprocessingml.document")
                        || contentType.equals("application/msword")) {
                System.out.println("Office 2007 or 2003!");

                // convert to PDF
                pdfRenditionFile =
ApiCmsImpl.convertDocToPDF(curRendition.getContentIS(),MimeMap.getExtension(fil
ename));
                Rendition pdfRendition = new Rendition();

        pdfRendition.setKey("/document/renditions/"+UUID.randomUUID().toString())
;
                //pdfRendition.setContentIS(pdfRenditionIS,);
                pdfRendition.setContentType("application/pdf");
                String newFilename =
filename.substring(0,filename.lastIndexOf("."));
                newFilename+=".pdf";
                pdfRendition.setFilename(newFilename);
                pdfRendition.setContentIS(new
FileInputStream(pdfRenditionFile),(int)pdfRenditionFile.length());
                recognizedFile = true;
                curDocument.setPdfRenditionKey(pdfRendition.getKey());

                em.persist(pdfRendition.getContentObject());
        }

        if(recognizedFile == true) {
                // generate images from first 10 pages
                List<File> resultPNGList = new ArrayList<File>();
                int pagesInPDF = ApiCmsImpl.numPagesInPDF(pdfRenditionFile);
                System.out.println("pages in pdf: "+pagesInPDF);
                for(int curPageNum=1; curPageNum<=pagesInPDF && curPageNum<=10;
curPageNum++) {
                        File conversionResultPNGFile =
ApiCmsImpl.convertPDFPageToPNG(pdfRenditionFile,curPageNum);
                        System.out.println(conversionResultPNGFile.getPath());
                        resultPNGList.add(conversionResultPNGFile);
                }
                String[] resultPNGArray = new String[resultPNGList.size()];
                int curPNG = 0;
                for(File curResultPNG: resultPNGList) {
                        Rendition curRenditionPNG = new Rendition();
                        File resizedPNG = ApiCmsImpl.resizeImage(curResultPNG,700,-
1);
                        curRenditionPNG.setContentIS((InputStream)new
FileInputStream(resizedPNG),(int)resizedPNG.length());
                        curRenditionPNG.setContentType("image/png");
                        curRenditionPNG.setFilename(curResultPNG.getPath());

        curRenditionPNG.setKey("/document/renditions/"+UUID.randomUUID().toString
());
                        em.persist(curRenditionPNG.getContentObject());
                        resultPNGArray[curPNG] = curRenditionPNG.getKey();

                        if(curPNG==0) {
                                // generate thumbprint
                                File thumbnailImage =
ApiCmsImpl.resizeImage(curResultPNG,150,150);
                                Rendition thumbnailRendition = new Rendition();
                                thumbnailRendition.setContentIS((InputStream)new
FileInputStream(thumbnailImage),(int)thumbnailImage.length());
                                thumbnailRendition.setContentType("image/png");

        thumbnailRendition.setFilename(thumbnailImage.getPath());

        thumbnailRendition.setKey("/document/renditions/"+UUID.randomUUID().toStr
ing());
                                em.persist(thumbnailRendition.getContentObject());

        curDocument.setThumbnailImage(thumbnailRendition.getKey());
                        }

                        curPNG++;
                }
                curDocument.setImageRenditionKeyList(resultPNGArray);
```

```
            // extract text from pdf
// isn't passing right extension        String extractedText =
ExtractText.extractText(new
FileInputStream(pdfRenditionFile),MimeMap.getExtension(pdfRenditionFile.getPath
()));
            String extractedText =
ApiCmsImpl.convertPDFToText(pdfRenditionFile);
            curDocument.setTextContent(extractedText);

        }

        // save document object
        ApiCmsImpl.setLifecycle(curDocument.getKey(),LifeCycle.PENDING_APPROVAL);
        Content contentObj = curDocument.getContentObject();
        new
ContentDAO(em,false).create(contentObj.getKey(),contentObj.getMetadata());

}
tx.commit();
System.out.println("Done!");
%>
```

Rendition.java

```
package document;

import java.io.InputStream;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.Map;

import entity.Content;

public class Rendition {
        String key;
        InputStream contentIS;
        int contentLength = -1;
        final public static String CONTENT_LENGTH = "content-length";
        String contentType;
        final public static String CONTENT_TYPE = "content-type";
        String filename;
        final public static String FILENAME = "filename";

        Content contentObject;

        public Rendition() {}

        public Rendition(Content renditionContentObject) {
                setContentObject(renditionContentObject);
        }

        private void setContentObject(Content contentObject) {
                if(contentObject==null) {
                        throw new RuntimeException("Rendition instance creation
failed: content object is null");
                }
                this.contentObject = contentObject;
                this.key = contentObject.getKey();
                this.contentType = contentObject.getMetadata().get(CONTENT_TYPE);
                this.filename = contentObject.getMetadata().get(FILENAME);
                String contentLengthString =
contentObject.getMetadata().get(CONTENT_LENGTH);
                if(contentLengthString!=null) {
                        this.contentLength = Integer.parseInt(contentLengthString);
                }
        }

        public Content getContentObject() {
                Content resultContent = new Content();
                if(contentIS!=null && contentLength!=-1)
                        resultContent.setContent(contentIS, contentLength);
                resultContent.setKey(key);
                resultContent.getMetadata().put(CONTENT_TYPE, contentType);
                resultContent.getMetadata().put(FILENAME,filename);
                if(contentLength>=0) {
                        String contentLengthString =
Integer.toString(contentLength);
                        resultContent.getMetadata().put(contentLengthString,
CONTENT_LENGTH);
                }

                return resultContent;
        }

        public String getKey() {
                return key;
        }

        public void setKey(String key) {
                this.key = key;
        }

        public InputStream getContentIS() {
                if(contentObject==null) {
                        return null;
                }
                InputStream contentIS = null;
                try {
                        contentIS = contentObject.getContent(); // call now to
support lazy loading
                } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                return contentIS;
        }

        public void setContentIS(InputStream contentIS, int contentLength) {
                this.contentIS = contentIS;
                this.contentLength = contentLength;
        }

        public int getContentLength() {
                return contentLength;
        }

        public String getContentType() {
                return contentType;
        }

        public void setContentType(String contentType) {
                this.contentType = contentType;
        }

        public String getFilename() {
                return filename;
        }

        public void setFilename(String filename) {
                this.filename = filename;
        }


}
```

ResizeImage.java

```
package processdocuments;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;


// convert %INPUT% -resize %WIDTH%x%HEIGHT% %OUTPUT%

public class ResizeImage {

        final static String CMD_ARGS[] = {
                "C:\\program files\\imagemagick-6.4.8-Q8\\convert", "%INPUT%", "-
resize", "%WIDTH%x%HEIGHT%", "%OUTPUT%"
        };


        /**
         * @param args
         */
        public static void main(String[] args) throws Exception {

                String filepath = "C:\\temp\\tiger-00005.png";
                resizePNG(new FileInputStream(filepath), 200, -1);

        }

        public static InputStream resizePNG(InputStream pngStream, int width, int
height) throws IOException {

                if(width<=0 && height<=0) {
                        throw new RuntimeException("width or height must be larger
than 0");
                }

                File inputTmpFile = File.createTempFile("resize_image-", ".png");
                File resultTmpFile = File.createTempFile("resize_image-", ".png");

                OutputStream inputOS = new FileOutputStream(inputTmpFile);
```

```
		byte[] buffer = new byte[1024];
		int bytesRead = 0;
		bytesRead = pngStream.read(buffer);
		while(bytesRead>0) {
			inputOS.write(buffer);
			bytesRead = pngStream.read(buffer);
		}
		inputOS.close();

		System.out.println(inputTmpFile);

		String[] cmd = new String[5];
		System.arraycopy(CMD_ARGS, 0, cmd, 0, 5);

		String widthStr = "";
		if(width>0) {
			widthStr = Integer.toString(width);
		}

		String heightStr = "";
		if(height>0) {
			heightStr = Integer.toString(height);
		}

		cmd[1] = cmd[1].replace("%INPUT%", inputTmpFile.getPath());
		cmd[3] = cmd[3].replace("%WIDTH%", widthStr);
		cmd[3] = cmd[3].replace("%HEIGHT%", heightStr);
		cmd[4] = cmd[4].replace("%OUTPUT%", resultTmpFile.getPath());

		Runtime runtime = Runtime.getRuntime();
		Process process = runtime.exec(cmd);
		try {
			process.waitFor();
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
//		InputStream is = process.getInputStream();

		System.out.println(resultTmpFile);

		return new FileInputStream(resultTmpFile.getPath());

	}

}
```

ReviewAction.java

```
<%
String key = request.getParameter("key");
String approval = request.getParameter("approval");
boolean approvalBool = false;
if(approval.equals("true")) {
	approvalBool = true;
}
```

```
EntityManager em = EMFactory.getEntityManger();
EntityTransaction tx = em.getTransaction();
tx.begin();
ContentDAO dao = new ContentDAO(em,false);

Document document = null;
if(key!=null) {
       Content content = ApiCmsImpl.getContent(key);
       document = new Document(content);
}


if(document!=null) {
       if(approvalBool) {
              document.setLifeCycleState(LifeCycle.PUBLISHED);
       } else {
              document.setLifeCycleState(LifeCycle.ARCHIVED);
       }
       dao.create(document.getContentObject().getKey(),document.getContentObject
().getMetadata());
}

tx.commit();
em.close();


%>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import="javax.persistence.EntityManager"%>
<%@page import="entity.EMFactory"%>
<%@page import="entity.ContentQuery"%>
<%@page import="entity.Content"%>
<%@page import="document.Document"%>
<%@page import="document.LifeCycle"%>
<%@page import="entity.ContentDAO"%>
<%@page import="javax.persistence.EntityTransaction"%>
<%@page import="api.ApiCmsImpl"%><html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>

Document: <%= key %><br />
Lifecycle Status: <% if(approvalBool==true) { %>Approved<% } else { %>Denied<%
} %><br />

</body>
</html>
```

search.jsp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>

<fieldset><legend>Basic Search</legend>
<form action="search_results.jsp">
<input type="text" name="query" size=80 />&nbsp;<input type="submit"
value="Search"/><br />
This field will search all text and metadata.
</form>
</fieldset>
<!--
<fieldset><legend>Advanced Search</legend>
</fieldset>
//-->
</body>
</html>
```

search_results.jsp

```
<%
String review = request.getParameter("review");
String query = "";
query += request.getParameter("query");
String actualQuery = "";
if(review!=null) {
	actualQuery = "life-cycle-state:pending-approval";
} else {
	actualQuery += "life-cycle-state:published "+query;
}

System.out.println("actual query: \""+actualQuery+"\"");
List<Content> contentList = ApiCmsImpl.search(actualQuery);
Highlighter highlighter = new Highlighter(new QueryScorer(new QueryParser("all-
text-and-attribute-values",new StandardAnalyzer()).parse(query)));
%>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import="org.apache.lucene.queryParser.QueryParser"%>
<%@page import="org.apache.lucene.analysis.standard.StandardAnalyzer"%>
<%@page import="entity.ContentQuery"%>
<%@page import="java.util.List"%>
<%@page import="entity.Content"%>
<%@page import="document.Document"%>
<%@page import="document.Attribute"%>
<%@page import="org.apache.lucene.search.highlight.Highlighter"%>
<%@page import="org.apache.lucene.search.highlight.QueryScorer"%>
```

```
<%@page import="org.apache.lucene.search.BooleanQuery"%>
<%@page import="api.ApiCmsImpl"%><html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>

<form>
<input type="text" name="query" value="<%= query %>" size="30"/>&nbsp;<input
type="submit" value="Search"/> <br />
<br />
</form>

<%= contentList.size() %> Result<% if(contentList.size()==0) { %>s<% } %><br />
<%
for(Content content: contentList) {
Document document=new Document(content);
String viewDocumentURL = "view_document.jsp?key="+document.getKey();
if(review!=null) {
	viewDocumentURL += "&review=1";
}
%>
<div style="border:3px white solid;float:left;width:100%;">
<a href="<%= viewDocumentURL %>" style="float:left;">
<img src="GetRendition?key=<%= document.getThumbnailImage() %>" border="1"
/><br />
</a>
<%
String[] hierarchyKeyList = document.getTypeHierarchyKeyList();
Attribute[] attributeList = new Attribute[0];
if(hierarchyKeyList.length>0) {
	String baseTypeKey = hierarchyKeyList[0];
	attributeList = document.getAttributeList();
	for(Attribute curAttribute: attributeList) {
		if(curAttribute.getTypeKey().equals(baseTypeKey)) {
	%>

<%= curAttribute.getName() %>: <%= curAttribute.getValue() %> <br />

<%
		}

	}

	String[] typeHierarchyNameList = document.getTypeHierarchyNameList();
	boolean first=true;
	%>Document Type: <%
	for(String  curTypeHierarchyName: typeHierarchyNameList) {
		if(first==false) {
			%>&raquo;<%
		} %>

		<%= curTypeHierarchyName %>

	<%
		first=false;
	}
}
%>
<%
String[] fragments = highlighter.getBestFragments(new StandardAnalyzer(),"all-
text-and-attribute-values",document.getTextContent(),5);
StringBuilder highlightedTextSB = new StringBuilder();
for(String curFragment: fragments) {

        highlightedTextSB.append(curFragment);
        highlightedTextSB.append(" ... ");

}
%>
<div style="clear:left;" ></div>
<% if(highlightedTextSB.toString().equals("")==false) { %>
<strong>Document Content:</strong> <%= highlightedTextSB.toString() %><br/>
<% } %>

<%
if(attributeList!=null) {
        for(Attribute curAttribute: attributeList) {
                String curHighlightedText = highlighter.getBestFragment(new
StandardAnalyzer(),"all-text-and-attribute-values",curAttribute.getValue());
                if(curHighlightedText!=null &&
curHighlightedText.equals("")==false) {
%>
                <strong><%= curAttribute.getName() %>:</strong> <%=
curHighlightedText %><br /><br/>
<%
                }
        }
}%>
</div>
<% } %>
</body>
</html>
```

select_type.jsp

```
<%!

public String printType(TypeTreeNode treeNode) throws IOException {
        StringBuilder sb = new StringBuilder();
        printType(sb,treeNode,0);
        return sb.toString();
}

public void printType(StringBuilder sb, TypeTreeNode treeNode, int level)
throws IOException {

        Type type = treeNode.getType();
        String typeName = type.getName();
        String typeKey = type.getKey();
        sb.append("<input type=\"radio\" name=\"type\"
value=\""+type.getKey()+"\" />");
        for(int i=0; i<level; i++) {
                System.out.print("&nbsp;&rarr;&nbsp;");
                sb.append("&nbsp;&rarr;&nbsp;");
        }
        System.out.println(typeName);
        sb.append(typeName);
```

```
       sb.append("<br/>\n");

       List<TypeTreeNode> childNodes =  treeNode.getChildren();
       for(TypeTreeNode curNode: childNodes) {
             printType(sb,curNode,level+1);
       }
}

%>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import="list.TypeTreeNode"%>
<%@page import="java.io.PrintWriter"%>
<%@page import="document.Type"%>
<%@page import="java.io.IOException"%>
<%@page import="list.TypeTree"%>
<%@page import="java.util.Set"%>
<%@page import="java.util.List"%><html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>
<form action="upload_file.jsp">
<%
TypeTreeNode rootTypeNode = TypeTree.getTree("/library/type/base");
%>
<%= printType(rootTypeNode) %>
<br/>
<input type="submit" />
</form>
</body>
</html>
```

StringUtils.java

```
package util;

public class StringUtils {

    public static String join(String[] s, String delimiter) {
       if(s==null||delimiter==null) {
             return "";
       }
        StringBuffer buffer = new StringBuffer();
        boolean first = true;
        for (String curS: s) {
             if (first==false) {
                    buffer.append(delimiter);
             }
            buffer.append(curS);
            first=false;
        }
        return buffer.toString();
    }
}
```

Type.java

```
package document;

import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import java.util.Set;

import util.StringUtils;

import entity.Content;

public class Type {
	private String key;
	public final static String KEY = "key";
	private String name;
	public final static String NAME = "name";
	private String parentKey;
	public final static String PARENT_KEY = "parent-key";
	public String[] typeAttributeUuidArray;
	public final static String TYPE_ATTRIBUTE_LIST = "type-attribute-list";
	private TypeAttribute[] typeAttributeArray;
	// see TypeAttribute for metadata prefixes

	public String getKey() {
		return key;
	}
	public void setKey(String key) {
		this.key = key;
	}

	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}

	public String getParentKey() {
		return parentKey;
	}
	public void setParentKey(String parentKey) {
		this.parentKey = parentKey;
	}
	public String[] getTypeAttributeUuidArray() {
		return typeAttributeUuidArray;
	}
	private void setTypeAttributeUuidArray(String[] typeAttributeUuidArray) {
		this.typeAttributeUuidArray = typeAttributeUuidArray;
	}
	public TypeAttribute[] getTypeAttributeArray() {
		return typeAttributeArray;
	}
	public void setTypeAttributeArray(TypeAttribute[] typeAttributeArray) {
```

```
                this.typeAttributeArray = typeAttributeArray;
        }
        public Type() {         }

        public Type(Content contentObject) {
                this.key = contentObject.getKey();
                Map<String,String> metadata = contentObject.getMetadata();

                this.name = metadata.get(NAME);
                this.parentKey = metadata.get(PARENT_KEY);
                if(metadata.get(TYPE_ATTRIBUTE_LIST).equals("")) {
                        this.typeAttributeUuidArray = new String[0];
                } else {
                        this.typeAttributeUuidArray =
metadata.get(TYPE_ATTRIBUTE_LIST).split("\t");
                }
                this.typeAttributeArray = generateTypeAttributeArray(metadata);
        }

        private TypeAttribute[] generateTypeAttributeArray(Map<String,String>
metadata) {
                Set<TypeAttribute> resultTypeAttributeSet = new
HashSet<TypeAttribute>();
                for(String uuid: typeAttributeUuidArray) {
                        String nameKey = TypeAttribute.ATTRIBUTE_NAME_PREFIX+uuid;
                        String typeKey = TypeAttribute.ATTRIBUTE_TYPE_PREFIX+uuid;
                        String optionsKey =
TypeAttribute.ATTRIBUTE_OPTIONS_PREFIX+uuid;
                        String name = metadata.get(nameKey);
                        String type = metadata.get(typeKey);
                        String options = metadata.get(optionsKey);
                        TypeAttribute typeAttribute = new
TypeAttribute(uuid,name,type,options);
                        resultTypeAttributeSet.add(typeAttribute);
                }
                return (TypeAttribute[])resultTypeAttributeSet.toArray(new
TypeAttribute[resultTypeAttributeSet.size()]);
        }

        public Content getContentObject() {

                Content resultContentObject = new Content();
                Map<String,String> metadata = new HashMap<String,String>();
                resultContentObject.setMetadata(metadata);

                //metadata.put(KEY,this.key);
                resultContentObject.setKey(this.key);

                metadata.put(NAME,this.name);
                metadata.put(PARENT_KEY,this.parentKey);
                //gen type attribute array
                String uuidStringList = "";
                boolean first = true;
                for(TypeAttribute curAttr: typeAttributeArray) {
                        if(first==false) {
                                uuidStringList+="\t";
                        }
                        uuidStringList+=curAttr.getAttributeUuid();
                        first=false;
                }
                metadata.put(TYPE_ATTRIBUTE_LIST,uuidStringList);

                putTypeAttributeArrayMetadata(typeAttributeArray, metadata);
                return resultContentObject;
        }

        public void putTypeAttributeArrayMetadata(TypeAttribute[]
typeAttributeArray, Map<String,String> metadata) {
                for(TypeAttribute currentTypeAttribute: typeAttributeArray) {
                        String currentUuid =
currentTypeAttribute.getAttributeUuid();
                        String currentName =
currentTypeAttribute.getAttributeName();
                        String currentOptions =
currentTypeAttribute.getAttributeOptions();
                        String currentType =
currentTypeAttribute.getAttributeType();


        metadata.put(TypeAttribute.ATTRIBUTE_NAME_PREFIX+currentUuid,
currentName);

        metadata.put(TypeAttribute.ATTRIBUTE_TYPE_PREFIX+currentUuid,
currentType);

        metadata.put(TypeAttribute.ATTRIBUTE_OPTIONS_PREFIX+currentUuid,
currentOptions);
                }
        }

}
```

TypeAttribute.java

```
package document;

public class TypeAttribute {

        final public static String TYPE_STRING = "type-string";
        final public static String TYPE_DATE = "type-date";
        final public static String TYPE_CHOICE = "type-choice";

        private String attributeUuid;
        final public static String ATTRIBUTE_STRING_PREFIX = "attribute-uuid-";
// then key
        private String attributeType;
        final public static String ATTRIBUTE_TYPE_PREFIX = "attribute-type-";
//then key
        private String attributeName;
        final public static String ATTRIBUTE_NAME_PREFIX = "attribute-name-"; //
then key
        private String attributeOptions;
        final public static String ATTRIBUTE_OPTIONS_PREFIX = "attribute-options-
"; // then key

        public TypeAttribute() { }

        public TypeAttribute(String uuid, String name, String type, String
options) {
                this.attributeUuid = uuid;
```

```
		attributeName = name;
		attributeType = type;
		attributeOptions = options;
	}

	public String getAttributeUuid() {
		return attributeUuid;
	}

	public void setAttributeUuid(String attributeUuid) {
		this.attributeUuid = attributeUuid;
	}

	public String getAttributeType() {
		return attributeType;
	}
	public void setAttributeType(String attributeType) {
		this.attributeType = attributeType;
	}
	public String getAttributeName() {
		return attributeName;
	}
	public void setAttributeName(String attributeName) {
		this.attributeName = attributeName;
	}
	public String getAttributeOptions() {
		return attributeOptions;
	}
	public void setAttributeOptions(String attributeOptions) {
		this.attributeOptions = attributeOptions;
	}

}
```

TypeTree.java

```
package list;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import api.ApiCmsImpl;

import document.Document;
import document.Type;
import entity.Content;
import entity.ContentQuery;

public class TypeTree {

	public static TypeTreeNode getTree(String nodeKey) {
		List<Content> root = ApiCmsImpl.search("key: "+nodeKey);
		Type rootType = new Type(root.get(0));
```

```
                TypeTreeNode rootNode = getTree(rootType);
                return rootNode;
        }

        public static TypeTreeNode getTree(Type rootType) {

                TypeTreeNode root = new TypeTreeNode(rootType);
                TypeTreeNode curNode = root;

                String curNodeKey = curNode.getType().getKey();
                List<TypeTreeNode> childrenTreeNodes = new
ArrayList<TypeTreeNode>();
                List<Content> childContentList =
ApiCmsImpl.getSubTypeChildren(curNodeKey);

                for(Content curChildContent: childContentList) {
                        Type curChildType = new Type(curChildContent);
                        TypeTreeNode curTreeNode = getTree(curChildType);
                        childrenTreeNodes.add(curTreeNode);
                }
                // sort list by name
                sortTypeTreeNodesByTypeName(childrenTreeNodes);

                curNode.setChildren(childrenTreeNodes);

                return curNode;
        }

        private static void sortTypeTreeNodesByTypeName(List<TypeTreeNode>
inputList) {
                System.out.println("sorting!");
                Collections.sort(inputList,new Comparator<TypeTreeNode>(){
                        @Override
                        public int compare(TypeTreeNode o1, TypeTreeNode o2) {
                                return
o1.getType().getName().compareToIgnoreCase(o2.getType().getName());
                        }
                });
        }


        public static void main(String args[]) { // for testing
                List<Content> root = ContentQuery.doQuery("parent-key:root");
                Type rootType = new Type(root.get(0));
                TypeTreeNode rootNode = getTree(rootType);
        }
}
```

TypeTreeNode.java

```
package list;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import document.Type;
```

```
public class TypeTreeNode {
	List<TypeTreeNode> children;
	Type type;

	public TypeTreeNode(Type type) {
		children = new ArrayList<TypeTreeNode>();
		this.type = type;
	}

	public List<TypeTreeNode> getChildren() {
		return children;
	}

	public void setChildren(List<TypeTreeNode> children) {
		this.children = children;
	}

	public Type getType() {
		return type;
	}

	public void setType(Type type) {
		this.type = type;
	}



}
```

TypeUtils.java

```
package entity;

import java.util.ArrayList;
import java.util.List;

import document.Type;

public class TypeUtils {
	public static List<Type> getTypeHierarchy(String typeKey) {
		List<Type> typeHierarchy = new ArrayList<Type>();
		String queryString = "key:"+typeKey;
		Content curTypeContent = ContentQuery.singleDoQuery(queryString);
		Type curType = new Type(curTypeContent);
		typeHierarchy.add(curType);
		while(curType.getParentKey().equals("root")==false) {
			curTypeContent = 
ContentQuery.singleDoQuery("key:"+curType.getParentKey());
			if(curTypeContent==null) {
				continue;
			}
			curType = new Type(curTypeContent);
			typeHierarchy.add(curType);
		}

		// reverse array elements so base class is first
		List<Type> tmpReverseArray = new ArrayList<Type>();
```

```
			for(int i=typeHierarchy.size()-1; i>=0; i--) {
				tmpReverseArray.add(typeHierarchy.get(i));
			}
			typeHierarchy = tmpReverseArray;
			return typeHierarchy;
		}
}
```

upload_file.java

```
<%
String typeKey = request.getParameter("type");
%>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import="entity.ContentQuery"%>
<%@page import="document.Type"%>
<%@page import="entity.Content"%>
<%@page import="java.util.List"%>
<%@page import="java.util.ArrayList"%>
<%@page import="document.TypeAttribute"%>
<%@page import="entity.TypeUtils"%><html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>

<%
List<Type> typeHierarchy = TypeUtils.getTypeHierarchy(typeKey);
%>

<form action="UploadFileServlet" method="post"  ENCTYPE="multipart/form-data">

<input type="file" name="file_upload" size="40"/> <br/>
<br/>
<%
for(Type hierarchyType: typeHierarchy) {
		String hierTypeName = hierarchyType.getName();
		TypeAttribute[] hierTypeAttributes =
hierarchyType.getTypeAttributeArray();

%>

<%= hierarchyType.getName() %> <br/>

<%
		String hierarchyTypeKey = hierarchyType.getKey();
		if(hierTypeAttributes.length==0) {
			%> <i>(No attributes)</i> <br/> <%
		}
		for(TypeAttribute curTypeAttr: hierTypeAttributes) {
			String curAttrUuid = curTypeAttr.getAttributeUuid();
			String curAttrName = curTypeAttr.getAttributeName();
```

```
              String curAttrType = curTypeAttr.getAttributeType();
              // parse options
              String curAttrOptions = curTypeAttr.getAttributeOptions();
              List<String> options = new ArrayList<String>();
              if(curAttrOptions!=null) {
                     String[] splitOptionsArray = curAttrOptions.split("\n");

                     for(String curSplitOption: splitOptionsArray) {
                            curSplitOption = curSplitOption.trim();
                            if(curSplitOption.equals("")) {
                                   continue;
                            }
                            options.add(curSplitOption);
                     }

              }
              String parameterName =
"attribute\t"+curAttrUuid+"\t"+curAttrName+"\t"+hierarchyTypeKey;
%>
              <%= curAttrName %>:

              <% if(curAttrType!=null && curAttrType.equals("text")) { %>
                     <input type="text" name="<%= parameterName %>" value=""/>
              <% } %>

              <% if(curAttrType!=null && curAttrType.equals("date")) { %>
                     <input type="text" name="<%= parameterName %>" value=""/>
              <% } %>

              <% if(curAttrType!=null && curAttrType.equals("choice")) { %>
                     <select name="<%= parameterName %>">
                     <% for(String option: options) { %>
                            <option value="<%= option %>" ><%= option %></option>
                     <% } %>
                     </select>
              <% } %>

              <br/>

<%
       }
       %>
       <br/>
       <%
}
%>
<input type="hidden" name="type_key" value="<%= typeKey %>"/>
<input type="submit" />
</form>
</body>
</html>
```

view_document.jsp

```
<%

String review = request.getParameter("review");
String key = request.getParameter("key");
Content contentObj = ContentQuery.singleDoQuery("key:"+key);
String[] imageRenditionKeyList = new String[0];
String[] typeHierarchyKeyList = new String[0];
String[] typeHierarchyNameList = new String[0];
Attribute[] attributeList = new Attribute[0];
Document document = null;
if(contentObj!=null) {

	document = new Document(contentObj);
	typeHierarchyKeyList = ApiCmsImpl.getTypeHierarchy(document.getKey());
	attributeList = document.getAttributeList();
	typeHierarchyNameList = document.getTypeHierarchyNameList();
	imageRenditionKeyList = document.getImageRenditionKeyList();
}

Rendition originalRendition = null;
Rendition pdfRendition = null;
if(document!=null) {
	String originalRenditionKey = document.getOriginalRenditionKey();
	String pdfRenditionKey = document.getPdfRenditionKey();

	if(originalRenditionKey!=null) {
		Content originalRenditionObj = 
ApiCmsImpl.getContent(originalRenditionKey);
		originalRendition = new Rendition(originalRenditionObj);

		if(pdfRenditionKey!=null && 
originalRenditionKey.equals(pdfRenditionKey)==false) {
			Content pdfRenditionObj = 
ApiCmsImpl.getContent(pdfRenditionKey);
			pdfRendition = new Rendition(pdfRenditionObj);
		}

	}

}

%>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" 
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import="entity.ContentQuery"%>
<%@page import="entity.Content"%>
<%@page import="document.Document"%>
<%@page import="java.util.List"%>
<%@page import="document.Attribute"%>
<%@page import="document.Rendition"%>
<%@page import="api.ApiCmsImpl"%><html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>

<div style="width:200px;float:left;">
<% for(int i=0; i<typeHierarchyKeyList.length; i++) {
	String curTypeHierarchyKey = typeHierarchyKeyList[i];
	String curTypeHierarchyName = "";
	if(typeHierarchyNameList.length>i) {
```

```
                curTypeHierarchyName = typeHierarchyNameList[i];
        }
        %>
        <strong><%= curTypeHierarchyName %></strong> <br />
        <%
        for(Attribute curAttribute: attributeList) {
                if(curAttribute.getTypeKey().equals(curTypeHierarchyKey)) {
%>
                <%= curAttribute.getName() %>: <%= curAttribute.getValue() %><br/>
<%
                }
        }
        %><br /><%
} %>
<% if(pdfRendition!=null) { %>
        Download PDF: <a href="GetRendition?key=<%= pdfRendition.getKey() %>"><%=
pdfRendition.getFilename() %></a><br />
<% } %>
Download Original: <a href="GetRendition?key=<%= originalRendition.getKey()
%>"><%= originalRendition.getFilename() %></a><br />
<br />
<% if(review!=null) { %>
<div style="width:90%;border:medium black solid;">
<strong>Review Panel:</strong><br />
<%
String reviewBaseURL = "ReviewAction.jsp?key="+key;
String approveURL = reviewBaseURL+"&approval=true";
String denyURL = reviewBaseURL + "&approval=false";
%>
<a href="<%= approveURL %>">Approve</a> <br />
<a href="<%= denyURL %>">Deny</a> <br />
</div>
<% } %>
</div>

<div style="float:left;">
<%
for(String curImageRenditionKey: imageRenditionKeyList) {%>
        <img src="GetRendition?key=<%= curImageRenditionKey %>" border="1" /><br
/>
        <br />
<% } %>
</div>

</body>
</html>
```