A Web Services Implementation Framework for
Financial Enterprise Content Management

Kevin H.S. Kwok and Dickson K.W. Chiu
Department of Computer Science and Engineering, The Chinese University of Hong Kong
e-mail: khskwok@yahoo.com, dicksonchiu@ieee.org

Abstract

There is an increasing demand to replace the current cost ineffective and bad time-to-market hardcopy publishing and delivery of content in the financial world. Four major goals, namely, management, cost, legal issues, and value, for a financial system should be reached. The new Financial Enterprise Content Management System (FECMS) could archive these goals by providing different content management, process establishment, and system integration functions. The main contribution of this paper is an enterprise content model and a system architecture for tackling two main technical challenges of FECMS, namely, global system integration and content flow management. It consists of three major types of engines: Content Editorial Engines, Content Reception Engines, and Content Publishing Engines. A Content Editorial Engine provides content and taxonomy maintenance and approval functions for different levels of administrators in the financial institute. A Content Reception Engine collects content from external sources, and then delivers it to different parts of the system for approval and publication. A Content Publishing Engine stores approved content and send them to different parties via different channels (such as email, fax, and conventional mail). It also serves as the Web storefront of the enterprise for user enquiries. In addition, we outline a unified scalable implementation framework with contemporary Web Services technology to support both internal content flow and inter-enterprise interactions. Both interactive users and institutional programmatic users are thus supported.

1. Introduction

Enterprise Content Management (ECM) refers to the management of textual and multimedia content across and between enterprises [31]. In the context of the financial industry, content refers to the pieces of information in the enterprise (in particular its Web sites), including financial research, market commentary, calendar events, trading ideas, bond offerings, and so on. Currently, there is an increasing demand to replace the current cost ineffective and bad time-to-market hardcopy delivery of content delivery and publishing in the financial world. Four major goals, namely, cost, management, legal issues, and value, for a financial system should be reached.

The management of such a large volume of content and such a complex system is non-trivial. For a global system with multiple sites, it is a big challenge to provide a mechanism for analysts all over the world to contribute commentary and publish them on the Web in a timely way. The maximum time to market of a commentary should be in minutes as its intrinsic value depreciates exponentially. However, an important contradicting requirement is that editors and auditors have to check content publication against possibility of violation of laws and regulations, which vary across countries and even states. Published content contributes highly to customer relationship management (CRM) as this is an important value-added service to clients in the financial industry, such as brokerage firms [7]. Content produced by analyst of a financial enterprise often provides valuable advice for decision making of client investors, and therefore has a high impact on the image and professionalism of the enterprise. In addition, content received or composed is also used throughout the enterprise for internal decision making. A good ECM system can produce high return on investment that is a valuable asset of the enterprise [22]. Thus, this is especially important for financial enterprises.

According to these requirements, the fundamental technical problems of a financial ECM system (FECMS) are global system integration and content flow management, both within and among enterprises. In summary, content is first received from various sources or composed in-house, then edited or commented, approved, and finally actively delivered to appropriate clients. To streamline even this basic content flow, intra- and inter-enterprise system integration is vital. Institutional clients often play the role of both content providers and content users and thus both incoming and outgoing content flow has to be automated with programmatic interface. However, currently, each enterprise has its own systems and interfaces. Intra-enterprise system integration (though seems more controllable) is also complicated. While maintaining glob-
ally consistent information (in particular content taxonomy), different departments and business units often in different geographical areas require separation of their sub-systems, because of regulatory, management, or technical reasons. The problem is further complicated by integration requirements with other enterprise information systems (EAI), especially legacy systems. Thus, a unified architecture and implementation framework is called for. The emerging Web Services technology [20] comes to the rescue, because Web Services provide loosely-coupled standard interfaces among autonomous systems within and among enterprises in the form of a set of well-defined functions for both programming and human user interfaces. In particular, Web Services support event-driven information integration for timely service provision and interaction [5][9].

Motivated by our recent study of a CRM system for Small and Medium Enterprises (SME) brokerage [7] that involves basic content delivery to customers, this paper explores a more in-depth research in a large-scale financial ECM context. The contribution and coverage of this paper are as follows: (i) requirements and technical problems of ECM in financial industry, (ii) a practical enterprise content model and architecture for such an environment, (iii) the design of FECMS components for effective and timely content flow management, and (iv) the use of Web Services for inter- and intra-enterprise FECMS integration. The rest of our paper is organized as follows. Section 2 introduces an overview of FECMS requirements and related work. Section 3 presents our enterprise content model and FECMS architecture. Section 4 presents the design and implementation framework of key FECMS components and the use of Web Services for system integration. Finally, we conclude our paper in Section 5 with discussion of some of our insights gained from this project and further research issues.

2. FECMS Overview and Related Work

First, we introduce some common terms used in a FECMS before discussing the main requirements for the stakeholders. Then we highlight management objectives and concerns. We also compare our approach with some related work.

Tagging refers to the labeling of content for easy classification, search, and retrieval. Tags can be thought of as index entries (meta-data) with specified values linked to a piece of content. All content are tagged when it is created. Some tags can be defined automatically by inference (for example, Country=China implies Region=Asia) or by templating, while others may need to be selected from a list of valid tags or specified by the author or editor. Templating refers to functionality for an individual to be able to save any particular piece of content information template for future use by the individual or the group.

Taxonomy refers to the overall structure and organization of tags across the enterprise. It is the basic mechanism for tiering, entitlement, and filtering of content. The taxonomy should reflect the creators’ view on what is important about any piece of content as well as the users’ view. In addition, it enables all content to be organized in a way that facilitates CRM activities, such as cross-selling, up-selling, and increase in customer orientation [30].

Different companies have different taxonomies. While the enterprise should maintain a consistent global repository of taxonomy, different business units may also have their own local taxonomies, say, because of language, terminologies, and regulatory difference. Some sort of mapping is required before delivery to different business units or external parties. For example, in securities’ world, product is regional/exchange base, such as Japan/Nikkei, US/NASDAQ/NYSE, Hong Kong/HKSE, and so on. But in the other business unit, products actually mean the financial institution provided instruments, such as Foreign Exchange Swap and Corporate Bonds. So, we have to re-map these tags to maintain the taxonomy ontology.

Entitlement is the ability to ensure that different types of customers and customers of different values are offered appropriate levels of service. Tiering is the ability to offer different levels of service (by providing access to different sets of content) to customers of different values.

2.1. Stakeholders Requirements and System Components

Based on a study of the FECMS of an international banking enterprise, Figure 1 depicts an overview of a FECMS, highlighting the main system components and stakeholders. A FECMS must be designed specifically to match the need and interest of each stakeholder within and related to the enterprise. Besides the management, there are four main types of stakeholders involved, namely, Content Creators, Content Providers, Content Distributors, and Content Users.

![Figure 1: Overview of a FECMS](image-url)
Content Creators collectively refer to internal users who involve in the content creation processes of the enterprise. The FECMS should be able to accommodate the different operational and administrative requirements of these different roles of internal users and to maintain appropriate security control. They interact mainly with Content Editorial Engines of the FECMS. Content Creators include the following roles.

- **Authors** compose content or publish content for analysts. They also provide initial tiering and tagging of the content. Content creation privilege is limited according to different roles. Different users can create different sets of content as classified by tags. Also, content flow is based on the user privilege and the content type of content. Some users (such as unit heads) may bypass the editorial or even the approval process but others cannot. Some content types allow straightforward processing but others may need multi-level approval. The system must be flexible enough to handle these variations in the content flow.

- **Editors** are power users to review content and tagging from authors or external sources and rectify it if necessary.

- **Approvers** review others’ content. All approvers are categorized by business unit, that is, content created by a certain business units requires approval from a particular group of approvers.

- **Auditors** review the content for the company interest and compliance to laws and regulations. Different from approvers who can only stop pending content, auditors can pull any piece of content back even if it has already been published.

- **Administrators** are super users to manage the overall operation of content creation. Administrators also maintain local or global taxonomy.

Content Providers are external sources (such as Reuters and Bloomberg) providing content (such as news, stock quotes, indices, and interest rates) to the enterprise through a Content Reception Engine. To ensure timeliness, content from trusted sources are usually forwarded automatically to the Content Publishing Engine for immediate delivery, relying on the tagging provided by the content source. However, editors and compliance auditors are able to review or withdraw them afterwards. On the other hand, content composed by the enterprise (such as market commentary, and research) is also delivered to these providers free of charge (public research), on per piece basis charge, or as a lump sum charge. This is because a major financial enterprise is usually also an important source of financial content.

Content Distributors are external service providers that render the content and delivery them to clients via different (traditional or electronic) channels, such as mass fax, mail, email, hardcopy delivery, and so on. Nowadays, these jobs are often outsourced. Though this is costly, additional services need to be maintained because of some clients’ needs and extra payment for them.

Content Users, who can be internal or external to the enterprise, are classified into five tiers in our case. In particular, content services to these external users are very important CRM activities. Content Users obtain content access through a Content Publishing Engine. They are maintained by an enterprise-wide Global Repository Management System. Based on their subscription data, the Content Publishing Engines also actively send appropriate content to the subscribed users. The five tiers are:

- **Public Visitors** – Anonymous users are often allowed to access some limited amount of public content through a portal. This helps attract them to visit the enterprise’s Web site.

- **Registered Visitors** – Potential customers who have not yet been using the enterprise’s services are attracted to register by the usefulness of the content. After registration, the enterprise knows more of the details of potential customers and therefore can perform more effective service recommendations and other marketing activities to them.

- **Clients** – Customers (such as, retail banking customers or SME) who do basic business with the enterprise are allowed full access and subscription to all unrestricted content. Their browsing and subscription provides further input to an analytical engine for mining opportunities for up-sale and cross-sale activities [30].

- **Priority Clients** – Premier customers (such as, private banking customers or institutional customers) who have deep relationships with the enterprise are allowed full access to all content that are not classified as “internal only”. Programmatic access of contents for institutional customers should be supported.

- **Internal Users** – Internal staff can access “internal only” content related to them, as well as all the content for external users. They are also automatically subscribed to relevant content, according to their job functions, market sector, geographical location, seniority, and so on. Based on similar criteria, further access control may be imposed.

### 2.2. Management Objectives and Concerns

To the management, the main goals of a FECMS are management, cost, legal issues, and value. Knowledge is power. As knowledge and organizational memory can be captured in enterprise content, access to content is an effective source of knowledge [21]. In addition, content distribution to clients is an important CRM activity, especially for financial enterprises. Thus, a FECMS provide a high value for both internal management and external marketing objectives. Besides the usual aim of an information system to reduce costs and improve management, a FECMS also help ensure compliance with legal issues and maximize value extracted from the content.
There is an increasing demand to replace the current cost ineffective and bad time-to-market hardcopy publishing and delivery of content in the financial world. Existing content management systems in many such enterprises are semi-manual and not integrated, mainly due to the diversity of external information sources, heterogeneous platforms, and legacy systems. Some of the sub-systems are redundant and not unified, that is, different sub-systems may exist in different units or in different geographical locations. Such chaos also results in high cost of system operations and maintenance. Thus, an integrated implementation framework for a new FECMS with standardized business processes and contemporary technologies is called for. Further, switching as far as possible to channels that are free at the point of delivery (such as electronic) reduces costs and unnecessary time delay.

At the same time, these problems also result in poor management of content and knowledge. To solve these management problems, standardized enterprise-wide policies and business processes provide a mechanism for content creation and management functions, such as content flow, document lifecycle, and collaboration. In particular, supporting different access control level as described in the previous sub-section is important to the integrity and security of enterprise content. On the other hand, similar to a library system, means for creating and maintaining the metadata (taxonomy) about the content is also vital for correct distribution and facilitates chances for further analysis. Further, there is a long-term need for integration with third-party FECMS or information sources as well as for defining and maintaining third-party content references.

Legal issues are important considerations because of the large amount of money involved and possible risks of damage to reputations. The FECMS should assist the management and Content Creators to ensure compliance with relevant laws and regulations. For example, the enterprise should adopt a single set of approval policy and procedures covering all forms of distribution in order to ensure that content are only published through official distribution channels (and not through personal mail distribution lists). The enterprise should also ensure that they know the identity of the Content Users, and what content they receive.

Besides timeliness, the FECMS should be able to deliver content across a wide range of clients (internal and external) and across multiple channels to increase its values in CRM. Further, if necessary, the editors should be able to edit the content and tailor it for a particular market sector. Finally, support for different formats of information, such as eXtended Markup Language (XML), Hyper-text Markup Language (HTML), Portable Document Format (PDF), graphics, image, multimedia, is essential for the currently Internet environment.

### 2.3. Literature Review

Enterprise Content Management (ECM) is an emerging research area. Tyrvänien et al. [31] give an excellent concise introduction to the research issues in this area, which mainly include technical, user, process, and content perspectives. McNay [22] presents an overview of ECM and stresses the need of an ECM system with consistent tagging to ensure a timely-updated, well-organized Web site. However, the paper does not cover any design of such an ECM system.

Croll et al. [13] point out that the trading of content between broadcasters requires descriptive data and some versions or illustrations of the content to be quickly assessed. The commitment should be confirmed and honored with minimal delay and administration, despite of complex content ownership and legal issues. Their Atman project attempts to model content trading using both archived programs and live events coverage as examples. Some of their requirements are similar to our FECMS but in a different application domain. However, available technologies nowadays can provide a much more sophisticated framework for similar applications.

Fensel [15] and Omelayenko [25] relate the challenges in inter-enterprise content management to B2B electronic commerce in the context of product information integration and ontology in electronic marketplaces. Kung et al. [21] relates knowledge management to enterprise Web content management with focus on superimposed information and domain ontology. They employ a Topic Maps approach in their system architecture because the underlying abstract model provides a high degree of power and flexibility to combine these approaches by supporting evolutionary construction of computer-based organizational memories. There are numerous researches in ontology in the context of Semantic Web [2] and therefore taxonomy ontology is not the focus of this paper.

Surjanto et al. [29] introduce XCoP (XML Content Repository) as a repository based on an object-relational database management system to improve content management of XML documents, thereby exploiting their structural information. Arnold-Moore et al. [1] describe the data model for implementing an XML-native content management server and the requirements for supporting text-intensive applications. However, these works present mainly technical details of a content repository. Weitzman et al. [33] present the Franklin Content Management System, developed by IBM's Internet Technology Group with XML technologies. Their goals are content reusability, simplified management of content and design that enforces integrity and consistency, the customization of content to individual users, and the delivery of content to a variety of display devices. However, multi-engine and heterogeneous engine integration issues essential for scalability and interoperability are not covered.
Chiu et al. [7] discuss the requirements of customer relationship management for SME stock brokerage in Hong Kong and propose an event driven approach to ensure efficiency and timeliness in converting knowledge into business actions effectively. One of such actions is to relay received stock price and market news content to relevant customers. This means ECM helps CRM. This motivates a more in-depth research on a large-scale ECM context, as presented in this paper.

Currently in the financial industry, wide-spread but scattered efforts of in-house development of ECM systems are emerging, because the value of FECMS has recently been appreciated. Such valuable solutions tailor-made for individual enterprises have been treated as trade secrets and therefore are rarely published. In summary, none of existing research papers has discussed a detailed enterprise-wide architecture that can adequately support a complete content flow, which includes content reception, creation, delivery, and management. Inter- and intra-enterprise ECM system integrations issues are almost unexploited.

3. Enterprise Content Model and FECMS Architecture

Based on the requirements discussed in Section 2, Figure 2 summarizes our Enterprise Content Model in Unified Model Language [23] class diagram, highlighting the main entities and their relationships in a FECMS. Figure 3 depicts our FECMS architecture in a highly heterogeneous environment with multiple data sources, external systems of business partners, and enterprise information systems (EISs). Web Services [20] allow e-commerce activities to be carried out based on a set of XML standards such as Simple Object Access Protocol (SOAP), Universal Description, Discovery and Integration (UDDI) and Web Services Description Language (WSDL). Traditional business-to-business (B2B) applications connect business partners through non-standardized architecture. A major drawback is the lack of interoperability, resulting in costly and time consuming connection establishment with a new trading partner can be. In contrast, the benefits of adopting Web Services include faster time to production, convergence of disparate business functionalities, a significant reduction in total cost of development and easy to deploy business applications for trading partners [26]. In addition, Web Services provide a convenient architecture to support both human and programmatic interfaces. Thus, external Content Providers, Content Distributors, and institutional clients are connected to the FECMS through Web Services.

At the same time, within the enterprise, Web Services can also provide loosely coupled communications among autonomous sub-systems. Multiple instances of each subsystem can be hosted at different sites (possibly in different geographical areas) for better management, performance, and scalability. Web Services support both synchronous and asynchronous messaging. Messaging provides a mechanism to support events, which can trigger immediate processing when relevant things happen, say, when new content arrives [7]. In particular, asynchronous messaging is heavily used to minimize the dependency among sub-systems and to provide reliable communications among different entities. Instead of traditional messaging services such as Java Messaging System (JMS) [28], Web Services provide an additional advantage of being an open platform that interoperates with heterogeneous systems and is firewall friendly [20].

![Figure 2. An enterprise content model in UML class diagram](image1)

![Figure 3. Architecture of a FECMS](image2)
unpublished content is much more isolated from Content Users. To further enhance security control, Content Reception Engines and Content Publishing Engines (accessed by both internal and external users and served as information gateways) are set up in a Demoralized Zone (DMZ) [27], with the protection of appropriate firewalls as shown in Figure 3. Content Editorial Engines and the Global Repository Management System contain valuable and sensitive data and are therefore set up within the intranet. It should be noted that there is only one Global Repository Management System in order to maintain users’ access to various global and regional Web sites as a single entity (compare airline companies’ portals), together with globally consistent content taxonomy.

UDDI registry support for Web Services helps in service location over the Internet as well as within the enterprise intranet. Content Reception Engines register their services in a public UDDI registry to allow external content providers to query its type of service and its availability. Similarly, Content Publishing Engines register their services in public UDDI registries to advertise the service of the enterprise, reaching new potential clients and improving access to current clients. They can also search for the appropriate Content Distributors through these registries to outsource conventional content delivery.

In a global enterprise, a large number of internal systems, services from various units possibly at geographically locations, and their interfaces are difficult to keep track of. A private UDDI registry can also serve for this purpose. Thus, UDDI technology helps manage and describe services as well as business processes programmatically in a single, open, and secure environment.

4. System Design and Implementation Framework

In this section, we outline our design and implementation framework of the main system components of the FECMS, including the Content Reception Engines, Content Editorial Engines, and Content Publishing Engines. We also focus on the interface design to demonstrate how Web Services effectively integrate them together and provide services to both internal and external Content Users.

4.1. Content Reception Engine

Figure 4 depicts our design of a Content Reception Engine, which is mainly for content reception and preprocessing. In order to process the received content and generated information effectively, we use an event-driven approach in the design, centered on an Active Rule Module. Our approach is motivated by the active database paradigm [6][14]. An *event* is the occurrence of something interesting to the system itself or to user applications. Receiving a content item (such as news or stock price update) is a typical event.

The Environment Listener receives input from Content Providers. Beside Web Services, we have to support a common message system, the Java Message Service (JMS) [28], such as for connection to Reuter’s services, until all information sources have been migrated to Web Services. Traditionally, information of Content Providers can only be provided by Content Reception Administrators. Some content sources require customized programming on proprietary protocols. Worse still, some even restrict content reception with special clients only, and therefore tricky wrapper programs are required for feeding them into the Content Reception Engine. With gradual migration to Web Services, a unified platform for reception can eventually be achieved. In addition, through public UDDI registries, service from new Content Providers can be sought for.

The Environment Listener mainly uses a publish-and-subscribe mechanism. Content Providers publish a summary of subscribe-able content at a public UDDI registry. This allows users to subscribe content that are relevant or interesting to them, through the subscription Web Service of the Content Provider. The Environment Listener’s corresponding reception Web service is provided to receive the subscribed content. Thus, polling Content Providers is not necessary. Instead, once when a new piece of content is ready, the Content Provider can actively send the content to the Environment Listener. Besides storing the received content into the Data and Content Warehouse, an event is generated to the Active Rule Module for processing.

The Data and Content Warehouse stores all information (including received content) and provides backing storage to the engine. The Analytic Module then analyzes the information in order to discover knowledge from the received content, such as summary reports calculations, market signal analysis, and so on. As discovered knowledge is very useful content, it is also stored in the Data and Content Warehouse and a corresponding event is also generated to the Active Rule Module for processing.
The rules processed by the Active Rule Module specify the constraints and actions to be taken upon reception and generation of new contents. It can be defined in the event-condition-action (ECA) format [14]. When an event occurs, it triggers some rules and the condition parts of these rules will be evaluated. Conditions are logical expressions defined upon content status and information, such as tags and their values. Only if the condition is evaluated to true, then the action part will be executed and may lead to other events. The semantics of ECA rules can be summarized by the following: On event if condition then action. As such, rules can be executed in a timely manner, avoiding the need of inefficient polling or ineffective batch processing. In addition, this provides a more flexible filtering and mapping capabilities than traditional table-driven approach. The ECA rules can serve for the following main purposes:

- Re-classify received content into the enterprise’s own taxonomy based on content information (such as tags and their values) as provided by the content source. Rule-based active filtering can be carried out too.
- Forward a selection of received or generated content to relevant analysts and Content Creators via different Content Editorial Engines for further analysis, editing, approval, and auditing.
- Forward selected content for immediate publishing via the Content Publishing Engine. Content from trusted Content Providers and some generated reports or signals are usually directly forwarded for publishing to keep the timeliness. Relevant Content Editors are also notified to continue the content flow.

We separate the Active Rule Module from the Analytical Module because the analytic engine should mainly deal with knowledge discovery, which tends to be resource intensive and computationally expensive. This cannot meet the timeliness requirements of urgent events for content processing. We also separate the Active Rule Module from other modules in order to manage the rules in a repository (the Content and Data Warehouse can also serve as the backing storage for the rules). This facilitates overall control and management of content sources for a particular Content Reception Engine.

### 4.2. Content Editorial Engine

Figure 5 depicts our design of a Content Editorial Engine. The engine’s function is to support Content Creators to create new content and work with received content. The operations of the engine are mainly alert driven, as timeliness requirement of financial content is crucial. Alerts are notification messages triggered by events and are managed by the Alert Management Module. The main function of the module is for alert routing and monitoring. Alerts further request the assigned user to carry out a job, with time (urgency) constraints [12]. If the default Content Editor responsible for the next step of a job in the content flow is not available or too busy, it will attempt to route the job and alert to a replacement Content Editor. It also monitors if the assigned Content Editor starts working on it and finishes the job within a deadline. Otherwise, reminder messages will be sent to the assigned Content Editor. Further internal mechanisms of an alert management system can be found in our other paper [12]. A typical creation content flow is as follows.

1. A Content Author creates a piece of content, determines its tier and tags, and then sends it to a Content Editor for revision.
2. Normally after editing, the content is approved by a Content Approver.
3. However, if the Content Editor suspects that the content might violate the laws or regulations of some countries, this piece of content will be sent to a Content Auditor of the affected country for compliance check. Before the Content Auditor’s approval, customers from those countries cannot receive or read it.
4. Once a job step is finished, the next one responsible in the content flow should be alerted to continue as soon as possible.
5. On the PC workstation of each Content Creator, special client software similar to ICQ [32]) is installed to receive the alerts. Further details of alerts and their associated jobs can be retrieved from the Content Editorial Portal in the form of a job-list.
6. By opening an entry in the job-list, the Content Editor by default acknowledges that work has been started.
7. Otherwise, he/she may decline the job (say, because of conflict of interest or specialty mismatch) by pressing the cancel button. If so, the Alert Management Module will try to find another suitable Content Creator for the job.

### 4.3. Content Publishing Engine

Figure 6 depicts our design of a Content Publishing Engine. The engine’s main function is to send new content to subscribed users and store them for later queries. In order to receive new content automatically, Content Users...
must subscribe to the relevant content categories beforehand. Interactive users may do this from the Content Access Portal, while this can also be done through programmatic Web Services interfaces. User registration and subscription are maintained with a Global Repository Management System so that users can interact with the enterprise as a single entity.

For example, different Hypertext Markup Language (HTML) outputs are generated for Web browsers on desktop PCs and Personal Digital Assistants (PDAs) respectively, while WAP Markup Language (WML) outputs are generated for mobile phones [19].

4.4. Global Repository Management System and Overall Integration

The Global Repository Management System provides backing support for user information and taxonomy. This includes all internal and external Content User registration, personal profiles, and their subscription data, in order to maintain users’ access to various global and regional Web sites as a single entity. In addition, the global taxonomy is maintained in this system for global consistency. The strategy is to keep minimal vital information in this repository to maintain its efficiency. Therefore, massive enterprise content is not stored here. In order to improve performance and reliability, replication techniques from contemporary relation databases, such as Oracle [16], may be used.

The overall system integration of the FECMS is based on Web Services interface to maintain autonomous sub-systems in various units of the enterprise. Figure 7 summarized the main Web Services offered by Content Reception Engines, Content Editorial Engines, Content Publishing Engines, the Global Repository Management System, and external Content Providers. Both inter- and intra-enterprise interactions are implemented with Web Services (labeled with external and internal, respectively). Let us examine a typical use case of content publish-and-subscribe through Web Services as follows.

1. An institutional user may request Web Services based content delivery by submitting a request to the updateSubscription Web Service of a Content Publishing Engine of the enterprise with the appropriate parameters, including the categories of required content and the address of its own reception Web Services access point.
2. The institution user has to implement a Web Service conforming to the specification of the receiveContent service of the Content Reception Engine.
3. The Content Publishing Engine verifies the request and relays successful request to the Global Repository Management System.
4. When new content arrives at the Content Publication Engine, the engine queries the Global Repository Management System through its getSubscribedUsers Web Service, with the tier and tags of the new content as parameters.
5. If the institutional user is included in the list, the Content Delivery Module of the Content Publication Engine will invoke the user-specified Web Service accordingly to deliver the piece of content.

Figure 6: Design of a Content Publishing Engine

Upon receiving a new piece of content, the Content Delivery Module queries the User Registration and Subscription System for the list of relevant Content Users with Web Services. The category of the content is determined by its tier and tags, which have been translated into the enterprise’s global taxonomy (see Section 4.1) or defined by the Content Creators (see Section 4.2). Only users subscribed to the category and with an adequately high tier are selected for delivery. In summary, the content is sent to the user via the following three main ways:

- Via email, SMS, and/or ICQ as specified by interactive users at subscription time.
- Via Web Services to the access point as specified by programmatic (usually institutional) users.
- Indirectly through external Content Distributors, usually for those who prefer traditional means (such as fax, or bulk mail). With Web Services, Content Distributors may be looked up via public UDDI registries and interacted automatically, without any human intervention.

Another main function of the engine is to support Content Users’ queries and browsing, which is mainly responsible by the Content Search Module. To maintain maximum reusability of code, all external access to the engine is performed through the Web Service Application Logic. Even the Content Access Portal has to invoke functions through these service points in order to maintain maximum code modularity. This approach is further justified because XML messages returned by Web Services can immediately be rendered with XML Stylesheet Language (XSL) technologies for users on different platforms.
Content Reception Engine  
External Service: receiveContent // for global taxonomy update  
Internal Service: receiveGlobalTag  
Internal Service: updateGlobalTag  

Content Editorial Engine  
Internal Service: receiveContent // for global taxonomy update  
Internal Service: addTag  
Internal Service: updateTag  

Content Publishing Engine  
Internal Service: publishContent  
Internal Service: checkDeliveryStatus  
Internal Service: updateContentStatus  
External Service: getSubscriptionCategories  
External Service: getSubscribedCategories  
External Service: updateSubscription  
External Service: getSubscribedUsers  
External Service: receiveContent  

Global Repository Management System  
Internal Service: createUser  
Internal Service: updateUserInformation  
Internal Service: getSubscriptionCategories  
Internal Service: getSubscribedCategories  
Internal Service: updateSubscription  
Internal Service: searchTags  
Internal Service: addTag  
Internal Service: updateTag  
Internal Service: addCategory  
Internal Service: updateCategory  
Internal Service: getSubscribedCategories  
Internal Service: getSubcriptionCategories  
Internal Service: updateUserInformation  

External Content Distributor  
External Service: deliverContent  
External Service: checkDeliveryStatus  

Figure 7: Design summary of Web Services interfaces

5. Discussion and Conclusion

This case study demonstrates that a complex ECM system can be decomposed into a set of highly coherent but loosely coupled sub-systems, which might be physically distributed within an enterprise. They are orchestrated by Web Services technology to work together seamlessly for the enterprise. This architecture is highly scalable and interoperable. There are no practical limitations in the implementation platform for each of these sub-systems as long as they support Web Services and programmed to compliant with the enterprise’s call conventions. For example, existing Java-2 Enterprise Edition (J2EE) enterprise applications can employ Sun Microsystems’ Web Service solutions [28] while current Content Editorial Engines using macros of Microsoft Office for its front-ends may be extended with the .NET framework [3]. For legacy systems, wrappers may be built around them to enable compatibility with Web Services. As such, upgraded subsystems can join the FECMS gradually for adequate testing and streamlining the switch-over, which might otherwise cause a great impact for a major enterprise-wide system.

In addition, Web Services serve as the middleware for interactions among business partners for sharing contents in both directions. Similar gradual migration strategies are also possible. In order to further streamline interactions among enterprises, application layer semantics (such as content taxonomy and category definitions based on Semantic Web research), protocols for interaction, and service-level standards are called for. Trade unions and regulatory bodies may help in such standardizations. If so, content service grids [17] can be formed for seamless and effective inter-enterprise content sharing and management.

We further consider the applicability of our ECM implementation framework to other industries is highly positive and optimistic. This is because of the trend in which organizations are moving towards service-oriented operations. For large enterprises, our operation model for ECM based on the anticipated content flow is highly generic. Though some enterprises currently do not consider importing contents from other sources, they eventually need to do so in order to increase their competitiveness. In addition, they will eventually realize the value of providing their clients with content for CRM. For SME, though the complete architecture may be overkill and too expensive, it is feasible to use some of these design concepts in their own architectures or software houses may develop packages with our approach. Moreover, the external Web Service interfaces presented in this paper is not complicated at all and can be easily programmed for content reception and delivery. This is because the main complexity (such as taxonomy and internal workflows) is encapsulated within the complete system but is not directly involved in the inter-enterprise interface.

The main extension and customization required for different enterprises to employ our framework is mainly in internal content flow modeling and role modeling of the personnel involved. In addition, more sophisticated security and access control mechanisms (such as role based or capability based approaches [11]) for content may be necessary, because tiering may not be adequate. This is also our main direction for future research. We would like to integrate the FECMS framework with a watermark infrastructure to reinforce document management policies by supporting non-repudiation in the document distribution protocol [4]. We are also investigating the application of an advanced workflow management system, such as ADOME-WFMS [10][11], in the FECMS for both inter- and intra-enterprise content flow management. In particular, we are interested in using the concept of flows in workflow based information integration and service negotiation [18].

In conclusion, we have presented a flexible, scalable, and sophisticated architecture for a Financial Enterprise Content Management System, based on a study of the requirements of a large international bank. We have proposed a unified scalable implementation framework with contemporary Web Services technology to support both internal content flow and external interactions. Four major goals, namely, management, cost, legal issues, and value, or a financial system have been considered during requirement elicitation. Our approach can archive these goals by providing effective content management, process establishment, and system integration functions. Further, we perceive that this pragmatic implementation framework is applicable to a wide range of enterprises.
References


