Overview
Document Framework
Version 1.0
December 12, 2005
Document History

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 5, 2005</td>
<td>Carl Yestrau</td>
<td>1.0</td>
<td>First complete version</td>
</tr>
</tbody>
</table>
Table of Contents

1.0 Introduction ...................................................................................................................................................... 1
2.0 Content Management History .......................................................................................................................... 1
3.0 Requirements for Managing Content .................................................................................................................. 1
   3.1 Creation of New Content ................................................................................................................................. 1
   3.2 Organization and Integration of Content ........................................................................................................ 1
   3.3 Intelligent Content Retrieval ........................................................................................................................ 2
   3.4 Content Transformation .................................................................................................................................. 2
4.0 Document Framework ...................................................................................................................................... 2
   4.1 XML Schema .................................................................................................................................................. 2
   4.2 XSL/XSLT ....................................................................................................................................................... 2
   4.3 WebDAV .......................................................................................................................................................... 3
   4.4 Unicode .......................................................................................................................................................... 4
5.0 Advanced XML Application Development (AXAD) ......................................................................................... 5
   5.1 Schema Modeling ........................................................................................................................................... 5
   5.2 Data Flow & Process Modeling ...................................................................................................................... 5
   5.3 Transformation Modeling ............................................................................................................................... 5
6.0 Conclusion ........................................................................................................................................................ 5
1.0 Introduction

A Document Framework provides universal access to information stored in any underlying database or proprietary content management system throughout a corporation. A Document Framework involves developing a loosely-coupled, distributed XML content management system based on open industry standards including XML Schema, XSLT, WebDAV and Unicode.

Benefits of implementing a corporate-wide document framework include reduced costs in creating, maintaining and distributing information across a variety of different media, and superior retrieval and sharing of information.

Success of a corporation is largely a function of their ability to cost-effectively capture, manage, publish and exchange its information assets, internally as well as externally, with customers and partners. A Document Framework provides the ability of leveraging existing database systems, current developer know-how, and open standards to produce a content and knowledge management solution tailored to the needs of an enterprise.

2.0 Content Management History

Many media types exist to deliver content. Web, wireless, web-services, e-books, DVD’s, print and new media types are being developed every year. All these media types require a process for creating, updating and delivering information to the end user.

In many organizations, corporate knowledge can be scattered across personal computers and a variety of server products including relational databases, XML repositories, properties files and proprietary content management systems. Corporations have a significant amount of resources invested in existing systems and it is worthwhile to adopt technologies that can be compatible with existing investments.

The above mentioned challenges can have potentially damaging effects on corporate productivity. In many situations there is no meaningful way of searching or retrieving information, resulting in loss or duplication of content. With multiple copies of similar data it is difficult to synchronize the maintenance of content which can result in outdated or erroneous data.

In an attempt to address the problems of managing content, many companies purchase proprietary document management systems. Often such document management systems have the inability to access enterprise data stored in database records.

Recently, new standards-based technologies have emerged which can potentially address the problems of creating and maintaining XML content. Additionally, cost reductions may be attained using these technologies that are referred to as Document Frameworks.

3.0 Requirements for Managing Content

A system attempting to address managing information across a corporation should consider the following four requirements; creation of new content, organization and integration of content, intelligent content retrieval and content transformation.

3.1 Creation of New Content

A system that can provide authors the ability to create content. The content authors ideally are not exposed to the underlying technical implementation of the system.

3.2 Organization and Integration of Content

A system that can provide saving functionality to any underlying storage facility. A storage facility may be a Relational Database, XML Repository or proprietary Content Management System. The system must further support concurrent editing and updating of documents, including versioning and source control options. Lastly, the system must provide content access
3.3 Intelligent Content Retrieval
Different database and content management systems can be optimized for information retrieval. A keyword search-engine versus a hierarchical yellow-pages directory listing both have strengths and weaknesses. A system must support enhanced content retrieval mechanisms while maintaining the context and structure of the data.

3.4 Content Transformation
Input-templates used to collect information, as well as the output stylesheets used to transform content to any output format (print, web, wireless, DVD’s, etc…)

4.0 Document Framework
A Document Framework can be considered a collection of software products and components used for developing a loosely-coupled system which manages content. The technological foundations of a document framework include XML Schema, XSLT, WebDAV and Unicode

4.1 XML Schema
XML Schema is a W3C official data definition language. XML Schema addresses many shortcomings associated with Document Type Definitions (DTD).

XML Schemas serve a critical role in the management of content as they can act as an intermediate translation bridge between databases and software objects. More specifically, XML Schema’s object orientated features allow for the mapping between XML document and objects written in any object-orientated programming language such as Java. XML Schema’s maintain data-type information and parent/child relationships, which allow for mapping of XML documents to other data stores.

![XML Schema Diagram]

Figure: XML Schema enabling translations from Objects to XML Documents and from XML Documents to databases (Interaction may be reversed).

4.2 XSL/XSLT
The eXtensible Stylesheet Language (XSL) and the eXtensible Stylesheet Language Transformations (XSLT) are standardized languages for transforming XML documents to a different output form. Using XSL, content saved in an XML format can be transformed into any output media through the applications of an XSLT stylesheet.

The eXtensible Stylesheet Language also includes a sub-set know as XSL:FO which stands for Formatting Objects. XSL:FO is an XML-based language for expressing production-quality document layouts used by many popular formats such as PDF and PostScript files.
4.3 WebDAV

WebDAV stands for Web-based Distributed Authoring and Versioning. WebDAV is a standardized set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web-server.
Implementations of WebDAV exist in many popular browsers, editors and servers. WebDAV enables the ability to transform how users interact with what is mostly now a read-only World Wide Web. Historically the Internet has been constrained to display and download capabilities WebDAV embedded in software enables to convert the Internet into a more writeable medium capable of supporting collaboration and distributed file sharing.

The protocol has features that include document locking and unlocking capabilities to prevent the overwriting of changes, versioning control and Secure Socket Layer (SSL) support for security.

It is important that a document framework implements a WebDAV repository interface to ensure compatibility with a wide range of server products. WebDAV technologies effectively enable a company to break free of vendor-lock-in-trap proprietary XML content management systems requiring that customers purchase both the content creation tools and back-end server products from the same vendor. Using WebDAV client software it is possible for content authors to collaboratively create & edit information stored in any WebDAV repository over the Internet.

Figure: Using WebDAV, content authors have distributed access to any underlying database or content management system.

4.4 Unicode

Unicode is a character encoding which provides a unique number for every character, regardless of platform, program or language being used. Encoding is of significant importance when dealing with Internationalization. Many content management systems do not readily support Unicode, or claim that they will support Unicode as a "new feature" can result in failed attempts to internationalize business systems.
5.0 Advanced XML Application Development (AXAD)

The Document Framework design process can be referred to as Advanced XML Application Development (AXAD). AXAD is an XML-centric design methodology which can be used for building Document Framework applications following a four-step process.

5.1 Schema Modeling

XML Documents require a content model, this is commonly expressed in the form of the W3C’s XML Schema. Developing an XML Schema is an iterative process which involves initial requirements analysis, use-cases, as well as examination of existing data schemas. Once the XML Schemas are designed, additional refinements are required to map all of the elements of your XML Schema to an underlying database (relational or XML-based) or content management system.

5.2 Data Flow & Process Modeling

The Document Framework involves content authors and content consumers. Content authors are non-technical domain experts whose primary task is to create content on any subject. Content consumers are typically customers, partners or internal departments who make use of XML content. The flow of information gathered by a Document Framework must be modeled with the following considerations:

- Who is the content author?
- What is the underlying data store?
- How is the information being transported to the database?

The above issues should be addressed during this stage.

5.3 Transformation Modeling

XSLT has a two-fold role in designing a document framework. XSLT is required to design both the input templates that are used by content creators, and the output stylesheets that are required by the content consumers. If the desired output is a printed document (e.g. PDF or PostScript file), then XSL:FO must be used. The developer must design the XSLT stylesheets to fit the data flow and process model determined earlier.

6.0 Conclusion

The business logic and user-interface of a Document Framework application can be custom developed. The components of a Document Framework consist of loosely coupled, lightweight components adhering to standard interfaces. A Document Framework can easily be implemented using any of the leading Internet application development platforms including J2EE, Microsoft .NET, Web-Services, Oracle Application Server, or most other platforms. Free XML processing Application Programmer Interfaces (API’s) are available for most programming languages, thus document frameworks can be implemented in Java, C#, C++, JavaScript, Perl, etc.