Transforming the Content Management Process at ibm.com

Louis Weitzman
Internet Technology Group, IBM
1 Rogers Street
Cambridge, Ma O2142
louisw@us.ibm.com

Sara Elo Dean
ibm.com, IBM
Laajalahdentie 23
00330 Helsinki, Finland
elodean@fi.ibm.com

Dikran Meliksetian
Internet Technology Group, IBM
150 Kettletown Road
Southbury, Ct 06488
meliksd1@us.ibm.com

Kapil Gupta
ibm.com, IBM
17 Skyline Drive, Bldg 2
Hawthorne, NY 10532
kapil@us.ibm.com

Nianjun Zhou
Internet Technology Group, IBM
150 Kettletown Road
Southbury, Ct 06488
jzhou@us.ibm.com

Jessica Wu
Internet Technology Group, IBM
150 Kettletown Road
Southbury, Ct 06488
jessicaw@us.ibm.com

Abstract

This case study explores the evolution of the Franklin Content Management System, developed by IBM’s Internet Technology Group. Franklin began as a technology-driven process to provide a web content management solution with the following goals: 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.

These goals were met in part by the decomposition of information into reusable fragments represented in XML. This approach provides unique opportunities in a content management system. However, it also raises some interesting challenges in the deployment of such a tool and the education of its users.

The development of Franklin evolved over a two year period and has culminated in the deployment of 62 country portals within the ibm.com domain. Furthermore, concepts from the Franklin project are influencing the strategy and design of IBM offerings.

Keywords

Software development, content management, content reuse, information architecture, XML, XSL, DTD, object dependency, web publishing, customization.

Industry Standards within Franklin:

XML: Extensible Markup Language (XML) is the universal format for structured documents and data on the Web. [1, 9]

XSL: Extensible Stylesheet Language (XSL) is a language for expressing stylesheets. It consists of three parts: XSL Transformations (XSLT), a language for transforming XML documents; the XML Path Language (XPath), an expression language used by XSLT to access or refer to parts of an XML document; and the XSL Formatting Objects (XSLFO), an XML vocabulary for specifying formatting semantics. [2, 10]

DTD: Document Type Definition (DTD) provides a means for defining the structure, content and semantics of XML documents. This standard has evolved into XML Schemas. [11]

HTTP: Hypertext Transfer Protocol (HTTP) is the protocol for communication on the web. [12]

WebDav: Web-based Distributed Authoring and Versioning (WebDav), is a set of extensions to the HTTP protocol to support users to collaboratively edit and manage files on remote web servers. [13]

Industry/category

Software development in the area of content management including information fragmentation, customization and delivery to multiple devices

Project statement

This project began as a technology-driven process for a new class of web content management system. It was motivated, in part, by past research [14], and the notion that designers cannot be present at all times in the dynamic environments in which we now live. Designers must create meta-designs, or representations of designs, to be used in the interactive and personalized delivery of information. This ability to provide "agile publishing" [5] challenges the standard practices in the design community. So this case study is not so much about an artifact, but rather, it is about supporting a new process of design.

This project was guided by the following principles: 1) the fragmentation of content into reusable document components, 2) the separation of the content from the design, 3) the conformance to industry standards, and 4) the presentation of a simple and understandable user interface. The result is Franklin, a unique web content management system now in use on all of IBM's country portals [6]. Over time, the project has evolved into a user-centered process as Franklin currently supports nearly 100 users worldwide.

The fragmentation of content in Franklin is supported by Trigger Monitor [3, 8], a core component of IBM's Olympic web site architecture. Trigger Monitor maintains the dependency of fragments within sites and automatically republishes any page when one of its component fragments gets modified.

The separation of content and design is supported by the industry standards of XML and XSL. In addition, other standards were used to comply with standard internet protocols for communication and representation (see sidebar).

A strong guiding principle in Franklin's development was to hide the details of the underlying data representations from the user. The solution provides a form-based interface for editing XML documents. This form is automatically created from DTDs and presents fields for only the data the user needs to edit. Other data is hidden or automatically generated by the system.

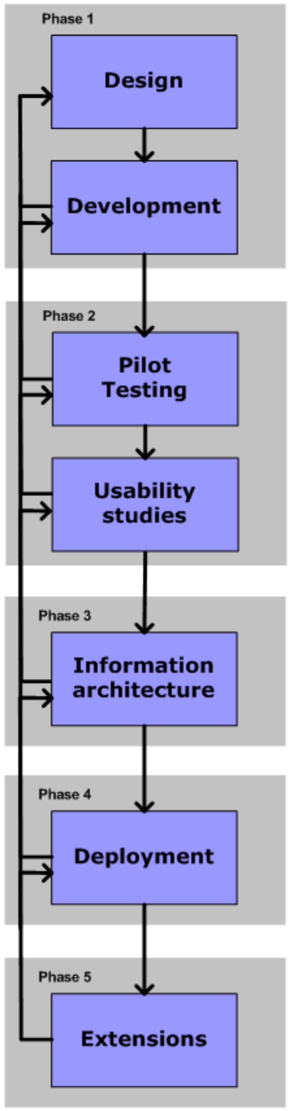
Project participants

This project relied on participants from a wide variety of disciplines. In order of appearance, members of the team and their roles: system architects created the initial software design, programmers and designers implemented the system, usability engineers helped identify problems with the user interface and the system, information architects designed the document types, stylesheet implementers created the style sheets, business developers built a business case and found potential users for the system, project managers oversaw the deployment, content editors migrated existing content and created new content with the tool, geography coordinators guided the deployment of country sites, and system administrators managed user profiles, system backups and ensured the system was up and running 7 by 24. These participants were part of a worldwide effort helping make the project a success.

Project dates and duration

The project began with the design of a prototype content management system in Fall 1999. By Fall 2000,

Figure 1. Project phases within Franklin



a working prototype had gone through initial pilots and two user studies. Further iterations improved the application, Then, in December 2000, ibm.com signed on to use Franklin in their portal customization project. Four initial countries were chosen as test cases to try out the system. These countries launched their portals in March of 2001. By January 2002, all 62 countries now publish their content with the Franklin system. Franklin manages over 13,000 document fragments. Additional work is ongoing for content customization and specialized delivery to different devices. Plans are in place to incorporate the ideas from Franklin into IBM offerings in 2002.

Design and development process

Franklin started as a technology vision of the Internet Technology Group. With the ibm.com engagement it evolved into a user-centered process. The development team began by building a prototype technology. Brainstorming lead to new features and problems were addressed as they arose. As Franklin reached an installed user base, the joint development and deployment team focused on user support through education, a problem tracking database, and updates to the tool.

Figure 1 illustrates the five project phases:

Phase 1: *Software design and development.* Focused on the building of a working prototype.

Phase 2: Pilot and usability. Evaluated the feasibility and usability of the prototype in the hands of users.

Phase 3: Information architecture. Defined the document types and corresponding stylesheets of the application.

Phase 4: Deployment. Rolled out the system worldwide.

Phase 5: Extensions. Enhances the system's capability to customize content to different audiences and to serve content to multiple device types.

Phase 1: Software design and development

The development team was spread across several cities on the East Coast of United States. In the design phase, this separation proved difficult and called for weekly face-to-face meetings. As the project progressed and the design solidified, working dispersed became easier. In fact, the physical separation helped to enforce the programmatic separation of the software modules of the system.

The software development phase continues today, as the development team rolls out new releases to meet the needs of the growing user base**.**

Phase 2: Pilot and usability

In the pilot and usability phase, content editors evaluated an early version of the client application implemented in Java. This helped the development team to better understand the content management process and how well Franklin supports this process. Since the content editors were brought into the design process early on, the development team was able to consider all feedback and integrate most of it immediately into the prototype.

The usability tests were conducted in a workshop setting totaling approximately 15 participants with 9 different scenarios. The workshops included:

1) an overview of the goals of the project,

2) a set of pre-workshop questions, to determine the background and job responsibilities of the evaluators so we knew which scenarios are most realistic for them and to get some initial reactions to the Franklin concept,

3) training scenarios, with related questions, to be completed by the participants in class,

4) homework scenarios, with related questions, that the participants could take home to work on at their own pace, and

5) final evaluation questions to be answered after all the exercises were completed.

The typical, initial response to the tool was that it seemed very complicated but after a couple hours working through the scenarios that view changed. Most users were able to perform all the tasks and actually found the tool to be easy to use. A final report was delivered that was a compilation of the scores for each of the various tasks and overall UI recommendations. Comments from the participants in the studies are described below. Users liked Franklin because it:

- Publishes data in as many different formats as desired
- Solves the problem of data maintenance on the web
- Stores data in XML
- Provides the ability to publish content without the help of developers
- Provides the ability to change content once and have the changes appear in multiple places
- Provides the ability to convert product data to non-web platforms
- Provides the ability to preview content
- Allows the sharing of fragments
- Provides better organization of content/data via standardization

Users' dislikes fell into two categories: Minor irritations that made the basic tasks more difficult, and major issues that were related to the new paradigm of content management. The lack of development time to implement a consistent user interface caused many of the initial minor complaints. Most of these issues have been addressed in later development. Some of the major issues and their solutions are listed below:

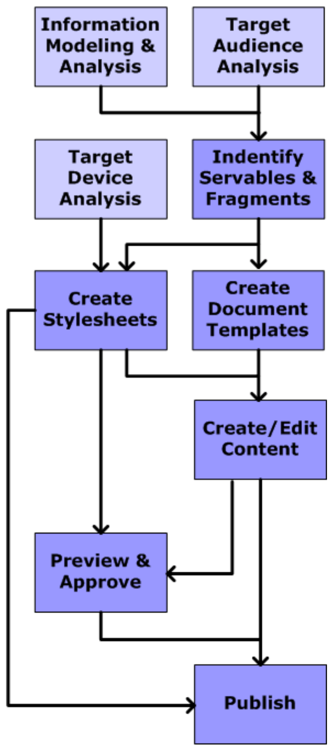
Conceptual model of Franklin. Web editors are accustomed to thinking of content as pages on a web site. A conceptual shift is necessary in order to understand the purpose and workings of a fragment-based approach to content management. A strong educational component was designed to make this shift easier.

Working offline. Content editors need to be able to work disconnected from the network. The Java standalone client was selected as the client interface for ibm.com. For working off-line, it is more appropriate than a web browser-based solution implemented in HTML and Javascript.

Code and data updates. The installation of the client application needs to be very simple. Code upgrades and bug fixes need to be distributed without significant impact on current users. A simple installer application and an automatic updater were created. The updater downloads any necessary code and DTD changes seamlessly at startup to each client application.

Finding documents. Content editors are accustomed to tools that represent a web site as a directory structure that can be browsed to find files. A new browse facility was incorporated into Franklin so that

Figure 2. Workflow for the deployment and use of Franklin.



users can explore documents on the server with a tree view without having to specify a search expression.

Support for multiple languages. IBM content editors author content in the languages of their country. Franklin was enhanced to display the user interface in the native language of the editor (including right-to-left display for Hebrew, and double-byte characters for Japanese, Chinese and Korean). After this feature was implemented, the requirement was relaxed to only display help strings in the native language and all other text in English.

User profile management. Two content editors may share the same workstation. In addition, passwords must expire according to the IBM security guidelines. Franklin was modified to enable multiple distinct profiles to coexist on a single client machine and to enforce the change of user password at regular intervals.

Phase 3: Information architecture

During the content definition phase, the Franklin development team and the ibm.com team developed the document type definitions (DTDs) and the stylesheets (XSLs) to model and present the content in Franklin. The overall process for deploying and using a Franklin-based system is shown in Figure 2.

All 62 countries and the 30 languages share the same document types, which include the home page, manual and automated index pages, news articles, contact pages, legal pages and general purpose pages. After considerable optimization to reuse data types as much as possible, 15 fragments and 12 servable document types were defined.

ibm.com mandates strict worldwide design standards, revised bi-annually. Although content guidelines exist as well, no rigid content standards have been previously enforced. Because each country had used their own content creation tools, no mechanism has existed to force pages to conform to a given structure. With the launch of Franklin, the pages were codified as DTDs for the first time.

After a number of iterations, the team focused in on the initial document types. As the project matured, however, it became clear that the initial set of sample documents had not accurately represented all the variations in the countries. For example, some countries use more than one contact name or a dateline in a news article. Therefore, the DTDs had to be adjusted as new countries rolled out. These modifications became more difficult over time because any change had to ensure that existing documents still complied with the updated DTDs. The tweaking of DTDs made it necessary to build tools to migrate content from the old definitions to the new definitions.

Stylesheets also followed an iterative design process. Since Franklin treats stylesheets as document fragments, any time a stylesheet changes all documents that rely on that stylesheet are updated. It was, therefore, very important to optimize stylesheet usage.

Initially, one stylesheet defined the layout for one document type with the elements specific to a county extracted into a separate stylesheet. As more countries adopted the system, common elements and exceptions became more obvious. When the structure of a document in a country was different from the norm, and would have resulted in an exception to the stylesheet, the country was asked to conform. Only in cases where the structure was judged to be of significant strategic value to the local circumstance was the exception allowed and encoded into the stylesheets.

During the roll-out, the stylesheets were restructured by extracting into separate stylesheets the common elements to all countries and languages, the common elements to a particular country and language, the common elements to a particular document type, and the unique elements to a particular document type in a country.

The significant effort of redesigning the stylesheets mid-stream paid for itself many times over. To accommodate a change today, only the relevant stylesheet is modified and thus only the corresponding output pages are automatically republished.

Phase 4: Deployment

The deployment phase consisted of a very aggressive schedule to train, migrate and launch all 62 portal countries in the ibm.com domain. This phase began with the selection and setup of the appropriate hardware infrastructure based on hardware sizing, scalability tests and the number of expected users.

The engagement with each country began with an executive letter to the general manager and the marketing and web management teams. Next, an initial conference call with the country web management and the ibm.com team ensured that any questions were answered in person. On the same call, the launch schedule was drafted and people identified to attend training.

The formal training materials were developed together with IBM Learning Services. Two components ensure appropriate focus: a two-hour management briefing informs country management about the benefits of the overall project, XML, and Franklin, and demonstrates the client application. The editor training includes an additional four hour hands-on exercise session which practices the typical content management tasks.

The first training session was conducted in person for editors from Canada, Chile, India and Switzerland. The subsequent sessions were “e-meetings”, a web based conference on the IBM Intranet that allows trainees to follow the trainer’s desktop through a web browser while following along the written course material and listening to a telephone conference call. This method enabled efficient worldwide training. Only in a few cases did poor network connectivity keep an editor from joining the e-meeting.

Migration of content began after training. The initial countries created between 20 and 150 fragments each. After launching 25% of the countries, the ibm.com team engaged migrators in order to keep up with the aggressive launch schedule. The editors were still responsible for the most visible sections of the site to ensure practice. In a few cases, due to absence or change in personnel, a country editor did not migrate any part of the site and thus did not acquire sufficient skills prior to launch. ibm.com has had to provide additional one-on-one post-launch training. Looking back, a country should have been required to acquire Franklin skills during migration so that once the site is live the content editors have the necessary skills.

During the deployment phase, the profile of those who interact with Franklin changed. Developers and testers were replaced by web editors, migrators, and administrators from ibm.com. The software tools also changed in this phase. Software modeling tools (Rational Rose) and application development environments (VisualAge for Java) were replaced with the Franklin user interface and XSL development tools, (XMLSpy and Emacs), and a problem tracking database (Lotus Notes).

Phase 5: Extensions

As the deployment phase of Franklin ends, ibm.com is defining the next features to roll out to the 62 country portals. The extensions planned for 2002 include content customization and delivery to wireless devices. While the Franklin design includes this functionality, the development team is working closely with the ibm.com team to ensure that the system supports the specific ibm.com requirements in these two areas.

Content customization has been piloted on the Sweden portal [7] starting in January 2002. The top-level pages present tailored content to users, who set their preferences for a country, language and area of interest.

The biggest implementation challenge was the extension of the existing DTDs and XSLs in such a way that un-customized and customized versions of documents can co-exist in Franklin. The customized content was modeled as a separate fragment and included in documents.

Another extension to Franklin will allow the delivery of ibm.com content to wireless devices. Again, the ibm.com DTDs and XSLs will be adjusted. DTD templates will be extended to accommodate content appropriate for the small display of cell phones and PDAs. For example, a content editor may input shortened abstracts of news articles or event announcements. New XSL stylesheets will produce various output formats such as WML and HDML. Since the design of Franklin allows multiple stylesheets to act upon existing content, this feature will require no system redesign.

Solution details

Two key decisions drove the design and extensibility of the Franklin prototype. The first was to embrace the industry standard XML. This means that all information within the system is stored as XML. DTDs define the allowed types of documents available to the users and XSL stylesheets transform those documents into the appropriate presentation format. In addition to serving as the data description language, XML is used as a means of communication between Franklin system modules.

The second decision was to create an architecture where components of information are reused throughout the system. Franklin defines the most basic documents as *fragments* which can be reused wherever they are needed. These *fragments* of information contain data as well as meta-data describing the fragment itself. *Servables* are fragments that contain additional information to produce one or more final published page.

To support this notion of information fragments, the Franklin architecture solves some of the basic problems this new approach creates. These problems include:

- How does a content editor search, find and reuse a previously created fragment based on its attributes?
- How is the dependency information between fragments maintained, and what happens to the dependents when a fragment is modified?
- How is the association between content and style maintained in the system and what happens when a style change is made?

- How is the complex internal structure of the XML fragments hidden from the content editors?

The Franklin system architecture consists of five interrelated components (Figure 3):

1. A *meta-store database* that stores the information about the documents,

2. A *dependency manager*, Trigger Monitor, that maintains dependencies between documents and invokes the necessary actions based on this relation,

3. The *file system* where the documents and assets are stored,

4. The *server* that coordinates the activities within the system, and

5. *Client applications* that allow users to create and edit the documents.

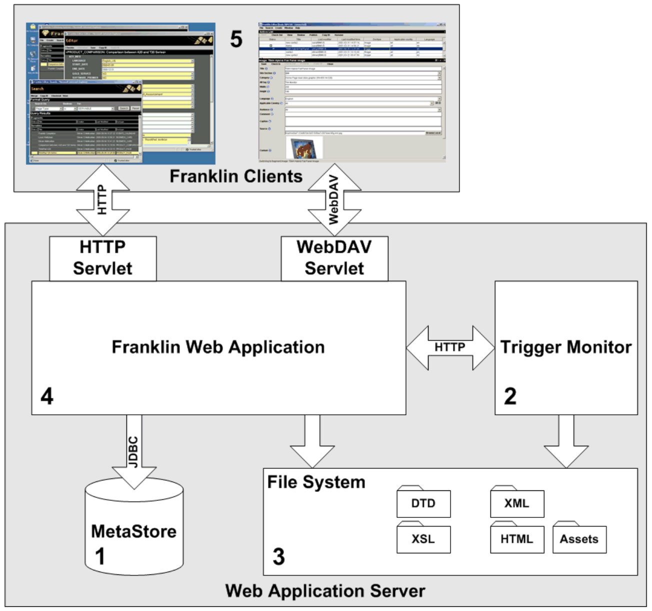


Figure 3. This diagram illustrates the 5 components of the Franklin system architecture.

1. Meta-store

The meta-store maintains information about the functional and semantic role of content fragments. It is a relational database (DB2/UDB) that maintains state information and enables fast searches.

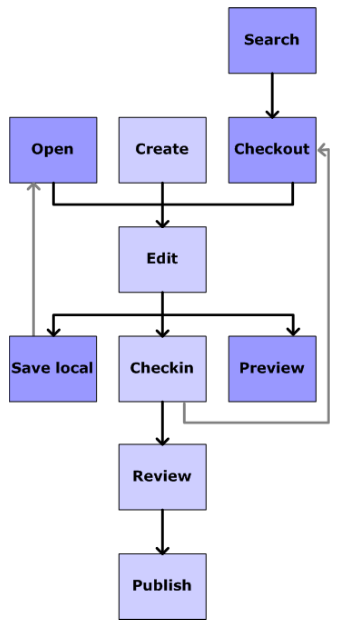
The meta-information of a document consists of system-generated elements, e.g. *document type* and *last modifier*, and elements filled out by the user, e.g. *title* and *summary* as well as meta tags such as *keywords* and *audience segment*. After a user edits a document and checks it into the system, the system automatically fills out the special system elements. The system then maps elements of the XML document to tables in the meta-store. The additional meta-tags are useful in customization as well as in search and retrieval of documents.

2. Fragment dependency store

The fragment dependency store builds upon Trigger Monitor, technology from IBM Watson Research [3, 8]. Trigger Monitor was originally designed to manage high numbers of rapidly changing HTML content fragments. By maintaining an object dependency graph, a directed acyclic graph, Trigger Monitor can efficiently update only the pages in an information space that need to be republished.

Trigger Monitor allows the use of specialized functions to perform tasks specific to a particular application. Using this feature, Franklin extends Trigger Monitor to maintain the dependency between XML fragments and

Figure 4. User interaction sequence within Franklin



the association between content and style. It also provides the methods to efficiently process the effects of a change in a content fragment or stylesheet.

3. File system

All documents are stored in the file system as XML files. Images, style sheets and other assets are stored in the file system as well.

4. Franklin server

The Franklin server manages the communication between the components of the system. When a document is checked into the system, it is added to the object dependency graph and a publish command is issued to the corresponding Trigger Monitor function. Multiple stages in the publishing process are possible. The ibm.com architecture uses two stages, one for preview and one for production.

5. Franklin client

One of Franklin's goals is to hide the complexity of the underlying XML representation from the content editor. This allows editors to create and edit content without worrying about the implementation of the underlying language. To achieve this goal, the Franklin client application presents a form-based interface to the user. In order to make the form sensitive to the type of XML element, the DTDs include attributes that indicate the type of interface component to display. For example, some elements are filled out by the system and are not displayed. Text elements entered by the user can be either single-line, multi-line or large text areas. Choice elements are presented as drop-down menus.

Because of the strict way the interface is constructed, the user interface knows when an element is required and highlights it appropriately. If a document is incomplete, the interface alerts the user and prevents checkin. Therefore, only well-formed and valid documents are submitted to the server to be processed.

Franklin provides access control through role assignment. Different roles provide access to different document types and publish directories. In addition, Franklin profiles can control language and country settings. Although Franklin does not support full workflow management internally, it has been hooked up to an external Lotus Notes-based workflow engine to control and sequence users' tasks.

The typical interaction sequence with Franklin is shown in Figure 4. The user can either create, checkout or open a locally saved document. This brings the document into the user interface to be modified. Once changes have been applied, the user can either preview without checking the document in, save it locally for offline editing, or check it into the system. The user must then review the document to verify the final layout of the fragment. Finally, the user publishes the document.

Authoring a news article in Franklin

A typical document in the IBM portal customization project is the news article. This document type has the largest number of instances within Franklin. This section illustrates how a content editor authors a typical news article. Figure 5 illustrates the end-to-end process of transforming DTDs into the final web pages. Figure 6 shows a news article being edited in Franklin. Figure 7 shows the resulting page generated from the news article, while Figures 8 and 9 depict a news index page and a home page that include partial content from the news article.

Authoring a news article in Franklin

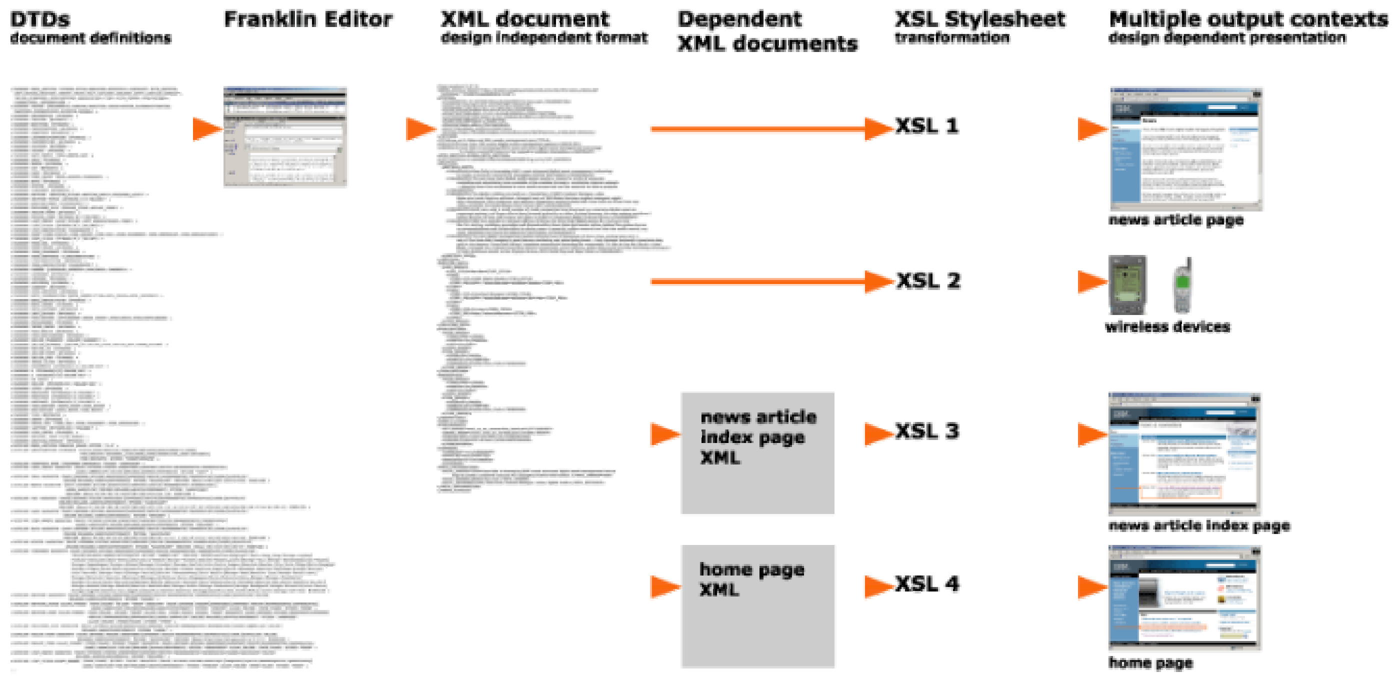


Figure 5. The process of creating a document in Franklin starts with the DTD and proceeds to the final presentations

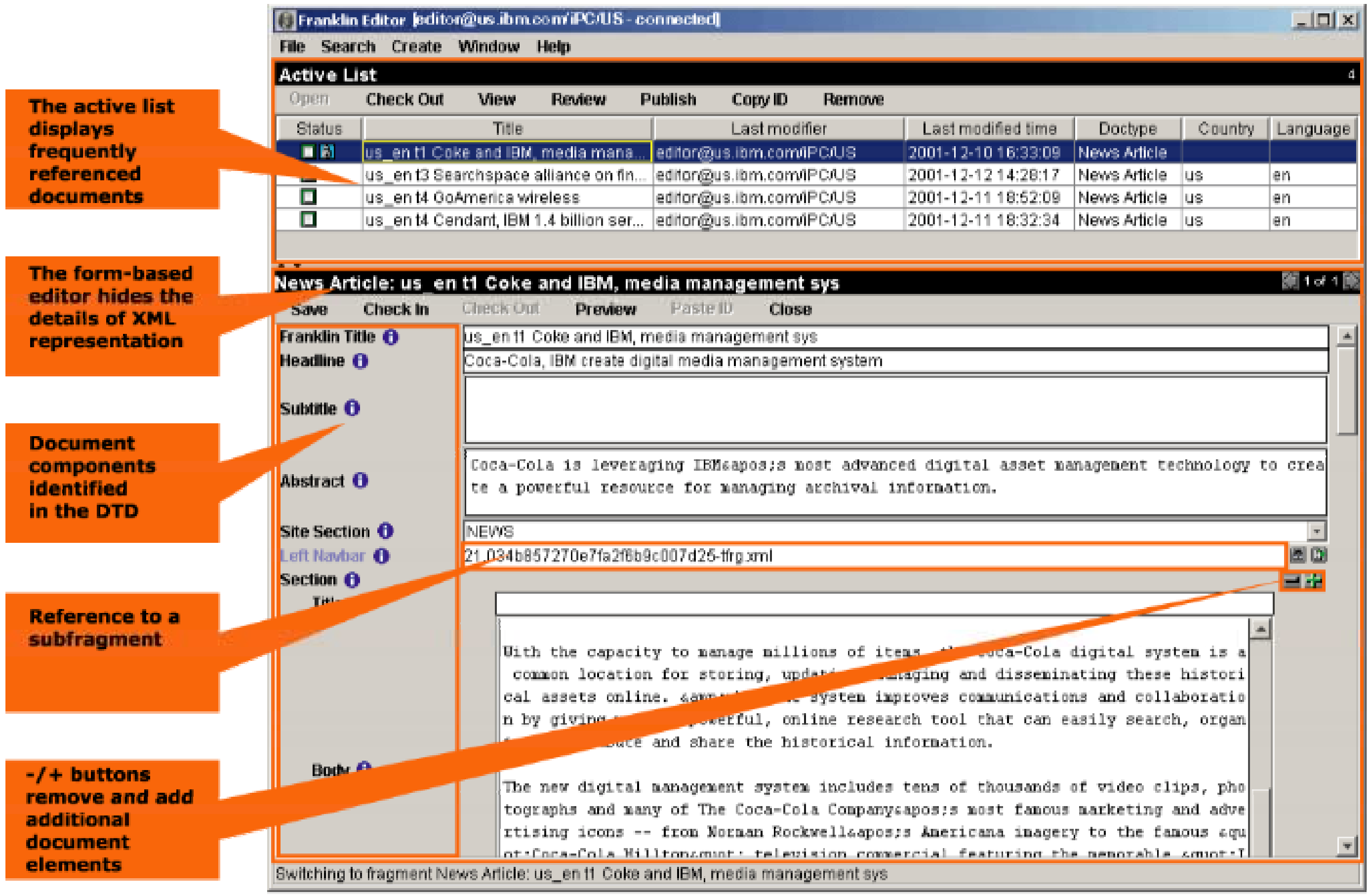
End-to-end process of creating a document

The creation of a document starts with the DTD. This is read into the Franklin client and converted into a form that the content editor can fill out. When the document is complete, it is checked into the system. An XML file is created and saved to the file system. Parts of the document are also indexed in the meta-data store so that it can later be searched. The XML document is then used as the starting point for a number of XSL stylesheets to create documents for different presentation contexts. On the web, a final news article page is produced as well as two other pages that news article contributes to, a news article index page and the home page with links to recent news articles. Meta-data queries generate these pages automatically. Additional pages may be produced for other devices.

Authoring a news article in Franklin (cont)

The Franklin Editor

The Franklin Editor is a Java standalone application. The application is divided into two panes. The upper pane lists documents currently or recently worked on. The lower pane is the input form where an individual document is created or modified. The DTD is translated into this simple form that the user can easily fill out. Here, a news article is being edited. In a DTD, elements can either be required or optional, and they can occur one or more, or zero or more times. If an element is required, the interface highlights it in yellow. If it can appear multiple times, -/+ buttons appear next to the widget to create additional copies, as seen in Figure 6.


The active list displays frequently referenced documents
The form-based editor hides the details of XML representation
Document components identified in the DTD
Reference to a subfragment
-/+ buttons remove and add additional document elements
Franklin Editor [editor@us.ibm.com/iPC/US - connected]
File
Search
Create
Window
Help
Active List
4
Open
Check Out
View
Review
Publish
Copy ID
Remove
News Article: us_en t1 Coke and IBM, media management sys
1 of 1
Save
Check In
Check Out
Preview
Paste ID
Close

Status	Title	Last modifier	Last modified time	Doctype	Country	Language
	us_en t1 Coke and IBM, media mana...	editor@us.ibm.com/iPC/US	2001-12-10 16:33:09	News Article		
	us_en t3 Searchspace alliance on fin...	editor@us.ibm.com/iPC/US	2001-12-12 14:28:17	News Article	us	en
	us_en t4 GoAmerica wireless	editor@us.ibm.com/iPC/US	2001-12-11 18:52:09	News Article	us	en
	us_en t4 Cendant, IBM 1.4 billion ser...	editor@us.ibm.com/iPC/US	2001-12-11 18:32:34	News Article	us	en

Franklin Title: us_en t1 Coke and IBM, media management sys

Headline: Coca-Cola, IBM create digital media management system

Subtitle:

Abstract: Coca-Cola is leveraging IBM&apos;s most advanced digital asset management technology to create a powerful resource for managing archival information.

Site Section: NEWS

Left Navbar: 21_034b857270e7fa2f6b9c007d25-frg.xml

Section

Title:

Body:

```
With the capacity to manage millions of items [illegible]Coca-Cola digital system is a
 common location for storing, updat[illegible]aging and disseminating these histori
cal assets online. [illegible] system improves communications and collaboratio
n by giving [illegible]werful, online research tool that can easily search, organ
[illegible]uce and share the historical information.

The new digital management system includes tens of thousands of video clips, pho
tographs and many of The Coca-Cola Company&apos;s most famous marketing and adve
rtising icons -- from Norman Rockwell&apos;s Americana imagery to the famous &qu
ot;Coca-Cola Hilltop&quot; television commercial featuring the memorable &quot;I
```

Switching to fragment News Article: us_en t1 Coke and IBM, media management sys

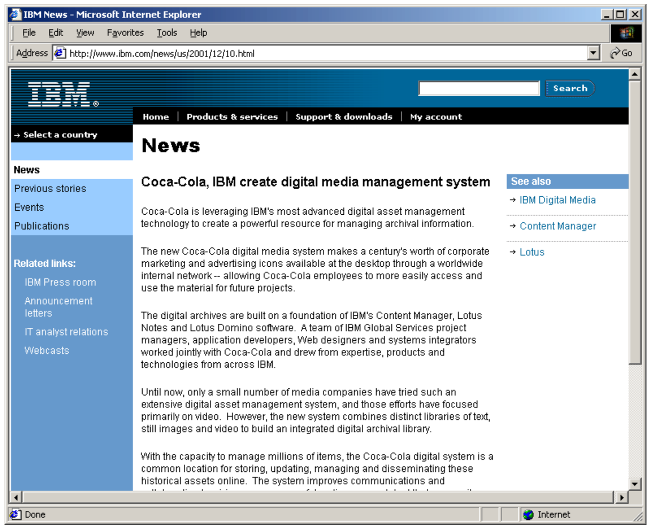
Figure 6. Form-based user interface automatically generated from document definitions (DTDs).

Authoring a news article in Franklin (cont)

News article pages

The XML news article is transformed into the HTML news article shown in Figure 7. This page includes a subfragment that describes the left navigation bar. Related links, on the right of the page, are included in the XML news article itself.

In addition, a pointer to this article is automatically placed on a news index page, in Figure 8, and on the home page, in Figure 9. A query to the meta-store finds the relevant news articles to include on these two pages. The relevance is determined by the type of news story as well as the publish and expiry dates entered by the editor.



News

Coca-Cola, IBM create digital media management system

Coca-Cola is leveraging IBM's most advanced digital asset management technology to create a powerful resource for managing archival information.

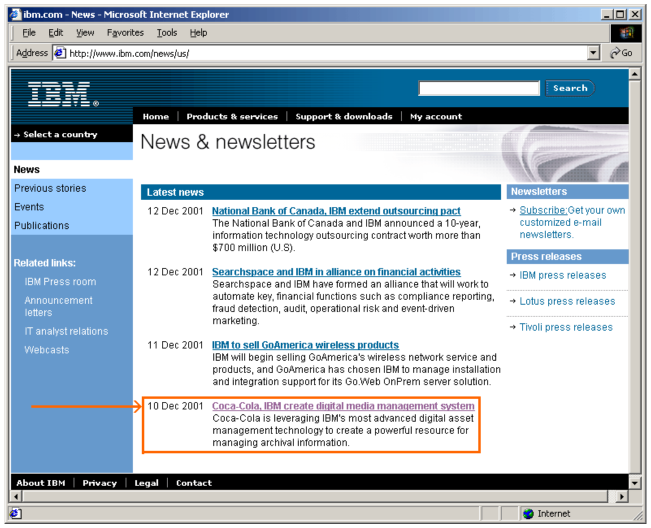
The new Coca-Cola digital media system makes a century's worth of corporate marketing and advertising icons available at the desktop through a worldwide internal network -- allowing Coca-Cola employees to more easily access and use the material for future projects.

The digital archives are built on a foundation of IBM's Content Manager, Lotus Notes and Lotus Domino software. A team of IBM Global Services project managers, application developers, Web designers and systems integrators worked jointly with Coca-Cola and drew from expertise, products and technologies from across IBM.

Until now, only a small number of media companies have tried such an extensive digital asset management system, and those efforts have focused primarily on video. However, the new system combines distinct libraries of text, still images and video to build an integrated digital archival library.

With the capacity to manage millions of items, the Coca-Cola digital system is a common location for storing, updating, managing and disseminating these historical assets online. The system improves communications and

Figure 7. The news article is presented on the live site.


ibm.com - News - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.ibm.com/news/us/
Go
IBM
Search
Home | Products & services | Support & downloads | My account
→ Select a country
News & newsletters
News
Previous stories
Events
Publications
Related links:
IBM Press room
Announcement letters
IT analyst relations
Webcasts
Newsletters
→ Subscribe: Get your own customized e-mail newsletters.
Press releases
→ IBM press releases
→ Lotus press releases
→ Tivoli press releases
About IBM | Privacy | Legal | Contact
Internet

Latest news

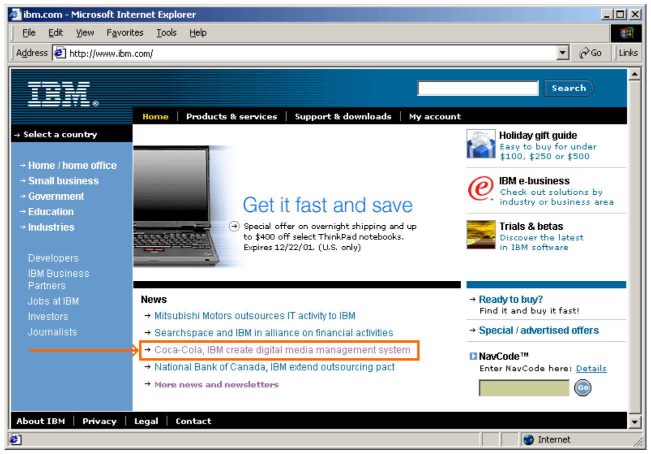
12 Dec 2001 National Bank of Canada, IBM extend outsourcing pact
The National Bank of Canada and IBM announced a 10-year, information technology outsourcing contract worth more than $700 million (U.S).

12 Dec 2001 Searchspace and IBM in alliance on financial activities
Searchspace and IBM have formed an alliance that will work to automate key, financial functions such as compliance reporting, fraud detection, audit, operational risk and event-driven marketing.

11 Dec 2001 IBM to sell GoAmerica wireless products
IBM will begin selling GoAmerica's wireless network service and products, and GoAmerica has chosen IBM to manage installation and integration support for its Go.Web OnPrem server solution.

10 Dec 2001 Coca-Cola, IBM create digital media management system
Coca-Cola is leveraging IBM's most advanced digital asset management technology to create a powerful resource for managing archival information.

Figure 8. The news article title, link and abstract are rendered automatically on the news index page.


ibm.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.ibm.com/
Go
Links
IBM
Search
Home | Products & services | Support & downloads | My account
→ Select a country
→ Home / home office
→ Small business
→ Government
→ Education
→ Industries
Developers
IBM Business Partners
Jobs at IBM
Investors
Journalists
Get it fast and save
Holiday gift guide
Easy to buy for under $100, $250 or $500
IBM e-business
Check out solutions by industry or business area
Trials & betas
Discover the latest in IBM software
→ Ready to buy?
Find it and buy it fast!
→ Special / advertised offers
NavCode™
Enter NavCode here: Details
Go
About IBM | Privacy | Legal | Contact
Internet

Special offer on overnight shipping and up to $400 off select ThinkPad notebooks. Expires 12/22/01. (U.S. only)

News
→ Mitsubishi Motors outsources IT activity to IBM
→ Searchspace and IBM in alliance on financial activities
→ Coca-Cola, IBM create digital media management system
→ National Bank of Canada, IBM extend outsourcing pact
→ More news and newsletters

Figure 9. The news article title and link are rendered automatically on the home page.

Benefits of Franklin

Franklin established a number of challenging goals at its inception. A powerful paradigm for content management results when merging the two main goals, of 1) the separation of content and design, and 2) the fragment-based document model.

The separation of content from design enables:

- Easy maintenance of content. Content is modified in one fragment and the update is automatically reflected in all relevant pages.
- Easy maintenance of design. Design rules are encoded and stored centrally which increases the integrity of the overall design strategy. It guarantees a strong and consistent brand across different sites.
- Easy delivery of content to new devices. The launch of new portals based on the same content is simple.

A fragment-based document model enables:

- Reuse of content within and across sites. There is no limit to the reuse of fragments.
- Intelligent update of modified content. When a piece of content is updated, all dependents of that content, and only those dependents, get automatically and efficiently updated.
- Intelligent update of modified designs. Designs are embodied in the stylesheets which are treated like any other fragment. When a stylesheet is updated, all content that relies upon that design is also automatically updated.
- Reuse of content for customization to audience segments. The fragments of information, when properly tagged, support customization of content to audience segments.

Technical accelerants

The Franklin system is built on industry standards wherever possible. In addition, the system relies heavily on Trigger Monitor. These technologies both enabled and constrained the implementation process. As the prototype progressed, the underlying industry standards also changed incorporating new programming interfaces. As a result, the system needed to be reengineered as it matured. For example, today XML Schemas are replacing DTDs as the prevalent document type description.

Trigger Monitor was originally designed for HTML and was not as efficient in handling XML objects. Franklin requirements drove innovation within Trigger Monitor to more efficiently handle XML internally. Franklin has motivated other projects within our group.

One technological hurdle was the mapping of XML into the meta-store. As a solution, the team devised a generic and efficient way to map XML to and from databases using industry standards transformation processing.

Post-launch results

The overall impact and impression of the Franklin prototype varies depending on the individual's perspective. From the point of view of the corporation, the roll-out has been very positive. All of the country portals are now using the tool and publishing their site using XML and XSL stylesheets. The site statistics as of January 2002 are:

- 62 total country portals have launched
- The resulting 66 websites comprise 5340 pages
- 13170 fragments exist in 30 languages
- 93 content editors worldwide add approximately 50 pages per day

Soon, the portals will be able to customize content to audience segments and publish to a number of different devices. In addition, cost savings have been realized in a number of different areas. These include:

- **Infrastructure and application hosting.** The content for all country portals is now hosted and supported on a single infrastructure instead of 52 infrastructures distributed worldwide.
- **Skill set reduction.** Franklin necessitates a simpler skill set for content creation. No special skills are required in the countries for creating content or deploying visual design updates, as those skills are now centralized within the ibm.com team.
- **More efficient content sharing.** Shared content is created once and used multiple times, rather than being copied for each site. The fragment-based content model is well-suited for content sharing by the globally

dispersed community of editors and will provide a foundation for further enhancements.

- **More efficient redesign.** ibm.com updates its site design approximately twice a year. Storing content in XML and rendering with centrally managed stylesheets significantly reduces the costs associated with this activity and enables tighter control over the design and overall user experience.

From the end user perspective, Franklin has radically changed the way the end users and the country web-masters work. Initially, they faced conceptual and motivational hurdles that needed to be overcome. When armed with proper education and hands-on experience, the users were able to overcome these initial difficulties. Most have come to appreciate the long-term benefits of modifying their past practices.

The new system has both reduced redundancy and created extra steps in the daily tasks. For example, the method of creating and updating news articles is now faster and more efficient, due to the form based interface, timed publishing and automatic archiving. Communication specialists do not have to rely upon web-masters to handcraft individual pages but can now upload the content directly.

On the other hand, the management of the country home page is now slower and more cumbersome. The home page DTD design decisions made in the information architecture phase were based on a sample of countries that were not fully representative. As a result, only a few countries can take advantage of the features in the home page management process.

The interface design has received mixed reviews from end users. On one hand, long and complex pages are now easier to update as the contents are presented in a form. On the other hand, HTML-savvy editors feel restricted by the fact that they can no longer control the layout of the page or create free-form HTML. In addition, because of constraints during development, the user interface does not offer the basic operations of most applications. For example, *find* and *undo* are not supported. These would make the interface easier to use.

Another inefficiency in the content management process could have been avoided with a better analysis of the team's organization in the various countries. Each country team should have been modeled as a group, with individuals having privileges to unlock or delete documents of other group members. Currently, no member of a country team can take these actions and have to request an administrator for help. End users find this aspect inefficient. On the other hand, the centralized infrastructure is more reliable, so publishing errors and delays have decreased since the migration. Overall, end users find they lose time in some steps of the new process while saving time in others.

One of the main design principles of Franklin, hiding the implementation details from the end users, was only partly achieved in the first prototype. For example, subfragments are sometimes displayed in the user interface with their internal fragment identifier, see Figure 6. Future versions of Franklin will better hide these details from a content editor.

Franklin, of course, is not suited for all web sites. It requires considerable server-side resources. In addition, the information modeling and definition of document types takes skill and time. Simple sites would not benefit from this approach. Franklin is best suited for large, frequently updated, structured content collections with a high degree of repetition, frequent redesigns, and multiple output formats.

Conclusion

With Franklin, ibm.com has established a sound foundation for an XML-driven content sharing model in which content and design standards are easier to deploy, manage, and update across all of IBM's country portals. Many similar efforts of business process re-engineering of content management have failed due to insufficient regard for the context, habits, and the goals of the end users [4]. While the Franklin system still has short-comings, the early inclusion of end users helped design the prototype to better support their tasks and improved acceptance. As Franklin evolves into the next phase, new goals have emerged. New system features and evolving industry standards will be considered for inclusion. In addition, with concepts from Franklin influencing future IBM offerings, the installed user base may soon be able to migrate to a fully-supported content management system.

Acknowledgements

We would like to thank the many participants that made Franklin and the Portal Customization deployment a success: members of the Internet Technology team past and present for helping create Franklin, the great group of technologists, designers and administrators at ibm.com for changing the way they work and supporting this new technology, and finally, the 93 content editors for using this prototype in their production environment and providing valuable feedback.

References

[1] Apache XML parser, Xerces http://xml.apache.org/xerces2-j/index.html

[2] Apache XSL processor, Xalan http://xml.apache.org/xalan-j/index.html

[3] Challenger, J., Dantzig, P., and Iyengar, A. "A Scalable and Highly Available System for Serving Dynamic Data at Frequently Accessed Web Sites" In *Proceedings of ACM/IEEE SC98*, November 1998.

[4] Cooper Interaction Design, November 2001 Newsletter http://www.cooper.com/newsletters/2001_11/whole_lotta_thwarting_going_on.htm

[5] Haimes, R. Managing Workflow and Content for Agile Publishing, Color Publishing, pp 24-33, January/February 1994.

[6] IBM country portals, http://www.ibm.com/planetwide/select/

[7] IBM Sweden country portal, http://www.ibm.com/se

[8] Iyengar, A., Challenger, J., Dias, D., and Dantzig, P. "Techniques for Designing High-Performance Web Sites," Submitted to *IEEE Journal of Internet Computing*, October, 1998.

[9] World Wide Web Consortium, Extensible Markup Language (XML), http://www.w3.org/XML/

[10] World Wide Web Consortium, Extensible Style Sheet Language (XSL), http://www.w3.org/Style/XSL/

[11] World Wide Web Consortium, XML Schemas and Document Type Definitions (DTD), http://www.w3.org/XML/Schema

[12] World Wide Web Consortium, Hypertext Transfer Protocol (HTTP), http://www.w3.org/Protocols/

[13] Web-based Distributed Authoring and Versioning, (WebDav), http://www.webdav.org/

[14] Weitzman, L. The Architecture of Information: Interpretation and presentation of information in dynamic environments. PhD Thesis, MIT Media Lab, February 1995. http://www.des1gn.com/papers/thesis.pdf