

# Tools and Methods for Learning Content Creation

Henry Meyerding

July 2003

## **Abstract**

One of the primary influences on the usefulness of an XML content repository are the tools used for creating content. There is much competition between software firms to secure a dominant position in the XML content market. Most of these tools are pitched toward documentation and not toward training. This document addresses these tools from a training perspective and discusses how they can be implemented into a plan for your XML repository of learning content.

# 1 Tools

This section analyzes some of the common tools that can be used with the XML content reuse repository. It is important to know how easy it is to construct new documents from content obtained from the XML database. It is equally important to know how easy it is to put new content created with this tool into the XML database so that it can be used in the future.

Anyone who grew up in the typewriter age might well be amazed at the layout, page formatting and document management capabilities of the current crop of software applications. As with all technology systems, there are prerequisites and "gotchas" – and at least three ways to do something:

1. The **RIGHT** way - the way originally envisioned by the developer and facilitated by the program.
2. The **WRONG** way - the way that someone found to make it work, because they didn't know what the right way was.
3. The **OTHER** way - the way that outwits the program and allows you to do something that should not be done, but needs doing.

Unfortunately for anyone who is facing the prospect of converting documents from various formats to XML, there is a considerable amount more **WRONG** and **OTHER** than there is **RIGHT** out there to be converted. Computers are infinitely stupid and must be told precisely what to do. In order for consistent content to result from an automated conversion to XML, consistent base content must be available. Consistency in the use of content creation applications is not a hallmark of most groups of instructional designers. Designers on a deadline are programmatic and care more about making it work now than about finding out how to make it work right later. It is paradoxical that a less intuitive tool, which requires more instruction and has a steeper learning curve, may be used more correctly and consistently than the naturally intuitive tool that everybody figures out for themselves.

## 1.1 Microsoft Word

Microsoft Word is the ubiquitous tool that does not play well with others. It has a long history of file format changes and inscrutable macros. Whether it can be used in conjunction with an XML content repository – and how well it can be used – comes down to two things: styles and templates.

To use Microsoft Word as an authoring tool is certainly possible. It is a fairly simple process to create an XSLT to convert XML content into a \*.doc or \*.rtf format so that it can be brought into Word. For example, if you are working with an XML document instance,

you can process that instance into an \*.rtf and send it to a reviewer who prefers to edit in Word. The problem happens when that review is returned to you and you wish to transfer those edits back into XML content.

Because users seldom use Microsoft Word properly<sup>1</sup>, it is seldom possible to convert Microsoft Word files to XML programmatically. Therefore, using Word decreases the productivity of the designers. Word does not operate in a manner consistent with structured documents. Using Word to author XML is like eating soup with a fork: you can do it, but it complicates things.

It is also true that practically every new version of Microsoft Word incorporates a plethora of undocumented changes in the file format. Changes in the format of the resulting Word files invalidates any programmatic automation that has been created. For this reason, most XML content systems use the more stable, but less capable \*.rtf format to transfer files to and from Word.

Many people consider that Microsoft Word has no place in an enterprise XML content reuse system. Some dedicated XML editors, such as Epic (see below) include filters to import Word content to XML.

Some other Microsoft programs, such as PowerPoint, can be used to create content and have very similar advantages and disadvantages to Word. Other Microsoft programs, such as Publisher or Front Page, pose another order of magnitude of difficulty in interoperating with content reuse systems.

## 1.2 Adobe and FrameMaker

Adobe FrameMaker is the WYSIWYG authoring tool of choice for XML applications. FrameMaker 7.0 includes a wealth of features that make authoring XML content much more efficient and practical. There are direct exports for both HTML and PDF document instances. Authoring in the structured view provides designers with an excellent means of understanding and using FrameMaker to create valid XML documents<sup>2</sup>

Adobe FrameMaker imports the XML data elements into a template. That template defines styles associated with the element definitions in the EDD<sup>3</sup>. This means that it is not

---

<sup>1</sup>People quite often learn to use Word by trial & error without instruction. They seldom know how to use templates or the styles they contain. When they want to have something in a different font or size, they apply that change from the tool bar, instead of applying a standard style to the text. Some Word users seem addicted to the space bar: instead of setting tabs appropriately, they achieve their indents through the use of multiple spaces. Word documents often contain revisions, highlighted text and complex section breaking. This kind of formatting makes programmatic chunking very difficult.

<sup>2</sup>Well-formed XML conforms to the syntax rules of XML: it is tagged correctly. Valid XML is well formed XML that conforms to the data structure defined in the Document Type Definition (DTD). All valid content is well-formed. Not all well-formed content is valid.

<sup>3</sup>the imported copy of the DTD

necessary to parse the XML and XSLT together to result in a formatted document instance. As the document is created, by adding structural components to the current document, the user sees the final format of their document.

Adobe FrameMaker can export files directly through Webworks to HTML. This is an option for training projects that rely extensively on interrelated print and online media. It can be easier to coordinate and publish the learning materials required if they are developed as a single source project. The base content is available from the XML repository, either as FrameMaker files or directly as XML.

Adobe FrameMaker also exports into Adobe Acrobat very well. Creating Acrobat files with FrameMaker allows you to include a lot of advanced Acrobat features<sup>4</sup> directly in the FrameMaker document, rather than having to modify the resulting PDF with Acrobat later. Creating Acrobat files with other programs, such as Microsoft Word, is much less efficient, unless the advanced features of the Acrobat format are not needed.

The downside of Adobe FrameMaker is that all this additional capability comes at a cost: it is not really very intuitive, especially for designers who are accustomed to work in Microsoft Word. It requires specialized technical expertise to set up correctly. Once it is set up, designers must be extensively trained in how to use FrameMaker properly. Many Word users are frustrated by the additional structure imposed by using XML. On the plus side, FrameMaker helps users to construct valid XML and informs them gently when their content is not valid. Of course, once they know that their content is not valid, they may need to have someone handy who really knows FrameMaker and its templates to help them to fix it.

Templates are the key. It is absolutely necessary to employ a dedicated Adobe FrameMaker expert to create templates. Most organizations do this on a consulting basis with one of the many Adobe/FrameMaker consulting firms.

Adobe has extensive training resources available, for a fee. They have a great deal of experience in implementing Adobe FrameMaker as an enterprise tool. If your organization makes the top-level commitment to pursuing an Adobe-enabled XML solution, the kind of support and expertise available from Adobe is unequalled elsewhere in the industry[2].

### 1.2.1 FrameMaker Server

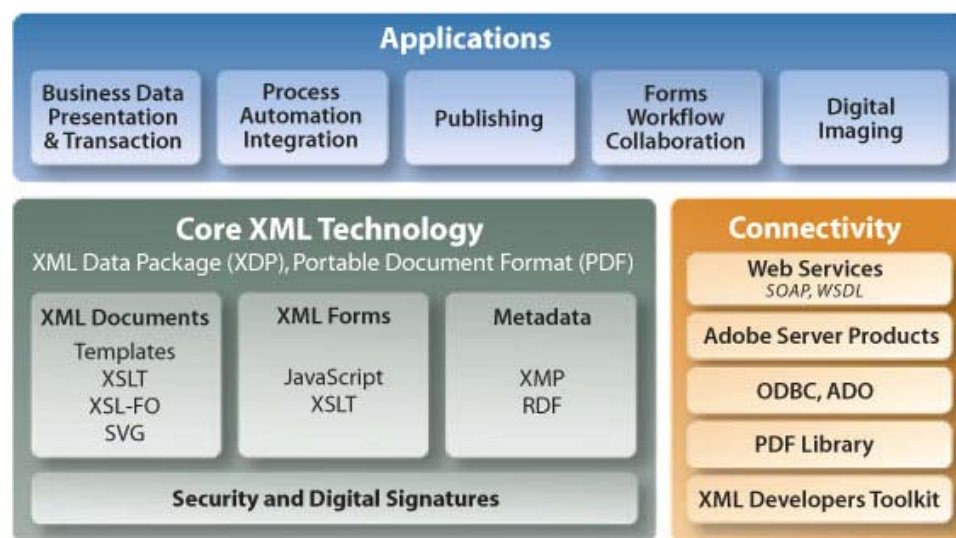
Adobe FrameMaker Server provides an opportunity to create a variety of dynamic documents. These documents, when accessed, perform real-time lookups of information from databases. That allows designers to access current information in a printable form. That ability is a great advantage for customer-facing training that requires frequent updates. It also could impact differential training, allowing designers to fill in the blanks with volatile information, instead of constantly trying to keep up with maintenance changes.

---

<sup>4</sup>Such as bookmarks, different kinds of linking, different security modes and so forth.

### 1.2.2 Adobe Document Server

Adobe Document Server[1] supports the dynamic creation of Adobe Acrobat documents from XML data. By flowing XML data retrieved from the XML database into document templates, you can generate instance documents and automated forms on demand. These document instances and forms can be highly complex, graphics rich presentations. Because they draw their content directly from the XML database, users always get the most current information. In addition, documents can draw upon multiple sources to populate document instances: XML content, PeopleSoft, SAP, LMS and other server content can combine in a single document instance that the user receives.



Adobe has many products and services designed for XML-based solution environments. Practically every application of XML technology has corresponding Adobe software designed for its implementation.

### 1.2.3 Adobe InDesign and GoLive

Adobe InDesign is a page-oriented<sup>5</sup> introduces built-in, extensible support for importing and exporting XML files. InDesign also allows you to export pages directly to Adobe GoLive 6.0 to use in dynamically generating Web pages. It supports SVG graphics and share native Photoshop and Illustrator files and can share these with GoLive. Through its tagged Adobe PDF support, InDesign exports graphically sophisticated eBooks that can be viewed on different devices. InDesign also supports Adobe Extensible Metadata Platform (XMP) for embedding metadata in documents.

---

<sup>5</sup>Page-oriented software allows a lot of flexibility and precision in placing content on each page; used for brochures and presentations. Document-oriented software is intended for larger, more complex documents where management of cross-references, indexes and other features is more important; used for manuals and other books.

Because it is a page oriented development tool, as opposed to content oriented, Adobe InDesign is a good choice for small 2-5 page documents where consistent look and feel is very important: marketing materials, offer briefs, and so forth. Many users find FrameMaker difficult to use in smaller, graphics intense documents. InDesign can be an excellent alternative.

Adobe GoLive is Adobe's competitor for Macromedia Dreamweaver. It does just about everything that Dreamweaver does, only a little differently. What it does not do as well as Dreamweaver is integrate as well with Authorware and Flash. If Lo-Fi<sup>6</sup> Web development is practically all your output, then GoLive may be an excellent choice, particularly if you are wishing to integrate more closely with print-deliverable development using FrameMaker. If Hi-Fi Web content is the majority of your online offering, then Dreamweaver has the edge in integrating with Flash and Authorware.

### 1.3 Macromedia Dreamweaver

For online content, Macromedia Dreamweaver is one of the most popular WYSIWYG HTML editors. Unfortunately, like Microsoft Word, it is often misused<sup>7</sup>. Templates are very important. Dreamweaver uses templates much in the way that FrameMaker does to add format to XML content. Dreamweaver imports XML into templates and generates HTML directly. Dreamweaver also exports XML content, which is efficient for people who like to work in HTML, but want the advantages of an XML repository.

**NOTE:** Dreamweaver does a good job of exporting the editable portions of templates as XML. However, it only checks whether the content is well-formed XML<sup>8</sup>, not whether it is valid XML.

HTML via Webworks-FrameMaker may not behave well in Macromedia Dreamweaver, since it has less tolerance of HTML code that it interprets as badly-formed XML<sup>9</sup>. It should be noted that using Adobe FrameMaker to write HTML results in Web sites that lack many of the features needed for richly interactive eLearning[3]. Webworks-FrameMaker works best for document based learning, where a large volume of information must be provided to the student as reference material.

---

<sup>6</sup>Lo-Fi Web content is primarily text with a few graphics and moderate interaction; suitable for thin client delivery. Hi-Fi Web content is highly graphical with strong user-interaction; video, Flash, and Authorware content are Hi-Fi.

<sup>7</sup>Many people learn Dreamweaver by using it, without any training. Like Word, Dreamweaver has many buttons and widgets that are convenient, but don't result in very good or consistent HTML. For example, someone may have extensive experience creating Web content with Dreamweaver but not have a clue about using templates.

<sup>8</sup>Well-formed XML conforms to the syntax rules of XML: it is tagged correctly. Valid XML is well formed XML that conforms to the data structure defined in the Document Type Definition (DTD). All valid content is well-formed. Not all well-formed content is valid.

<sup>9</sup>HTML is well formed when it conforms to the syntax of the version of HTML supplied in the document definition



The big advantage of Macromedia Dreamweaver is that many people feel comfortable with it. It is another learning step, but a relatively easy one to understand how to import and export XML in Dreamweaver. Again, it is of paramount importance that the templates into which XML is imported are used verbatim. It is a very good idea to have those templates generated by expert consultants, if sufficient Dreamweaver specific expertise does not exist in your organization.

### 1.3.1 Authorware and Flash

It is perfectly possible to create learning objects in Flash or Authorware and store them in the XML repository. It is usually a good idea to break up longer Flash and Authorware segments into scenes. In this way you can reuse particular content without having to modify a large, complicated segment when only part of it is desired.

## 1.4 Arbortext Epic Editor

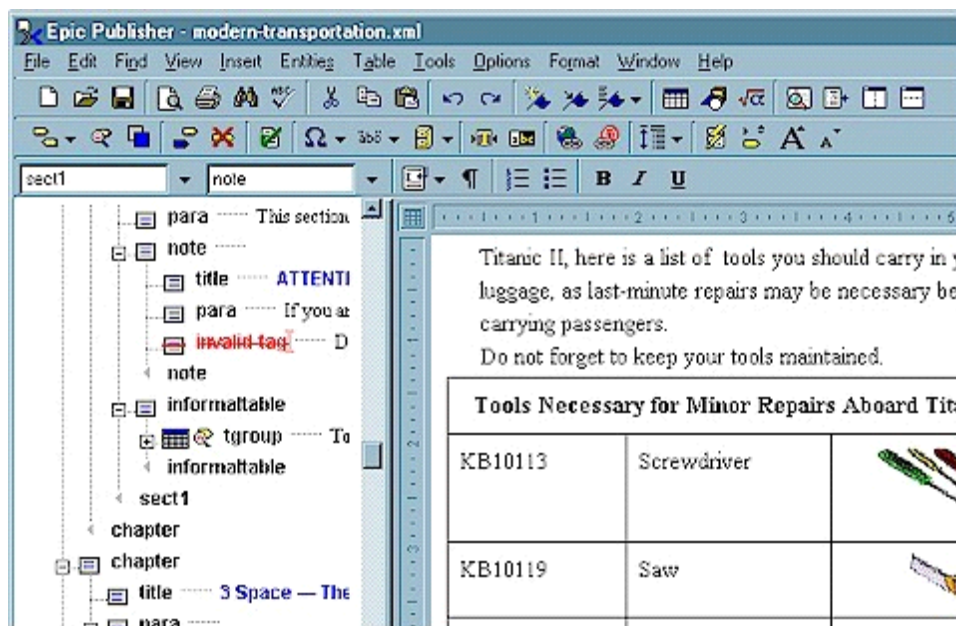
Arbortext Epic Editor is a purpose-built XML editor. It handles a broad range of applications and does a good job of providing an editing interface for XML content. The user interface is good, but not at all like the standard WYSIWYG document editing environment. Like Adobe FrameMaker, it is a groupware product that is specifically optimized to handle:

- Content collaboratively written and maintained by teams of authors working in multiple languages.
- Content created in reusable components independent of their formatting, stored in content management repositories, and dynamically assembled on demand.
- Content personalized for specific audiences and formatted for delivery on multiple media: Web, CD-ROM, print and wireless.
- Content automation based on systems and software that are easily customized and leverage the broadest available support for XML and related standards.
- Content creation through client-based installations for occasionally disconnected users and through server-based installations accessed by Web browsers for users who are connected full-time.

Out of the box, Arbortext Epic Editor works with file systems, WebDAV-enabled repositories, and has configurable adapters for Documentum, Oracle CM SDK (formerly named iFS), and FileNet Panagon Content Services. Arbortext's other repository partners provide adapters to Epic Editor, including BroadVision One-To-One Enterprise, empolis SigmaLink, Progressive Information Technologies Target 2000, and XyEnterprise Content.

Arbortext offers separate products for content conversion and publishing. The Enterprise E-Content Engine (E3) converts content from Microsoft Word, Adobe FrameMaker and Interleaf documents to XML, and publishes dynamic content to print / PDF and Web / wireless. To publish to CD-ROM, Arbortext offers the CD-ROM Composer.

This E-content engine is an off-the-shelf parser that can be used to automate many different kinds of legacy chunking operations. It does not work miracles: Nothing will correctly parse badly formatted Microsoft Word files (that requires human intervention and exercise of good judgment). It does provide to the enterprise a tool the equal of, or better than, many learning content parsers that typically require a much higher investment for the same return.



Arbortext Epic Editor is the best of a series of content editors that have attempted to get the most out of XML structure, which allowing users to see a visual representation of their output. Given that the designer understands XML and the learning content, Epic can run circles around FrameMaker as a tool for importing and creating new content. The Epic editor is very often imitated by LCMS vendors that work in structured document formats.

## 1.5 Corel XMetaL

Corel XMetaL is part of a suite of XML applications. It is an advanced structured editor that is relatively easy to use and highly customizable for applications based on well-known DTDs. It provides three views of an XML document: a plain text view in which you can view the underlying XML code; a tags-on view in which elements are represented as symbols in a formatted document; and a normal view that displays the formatted doc-



ument and hides the markup. XMetaL supports use of Cascading Style Sheets (CSS) to control the formatted view of the document on screen.

Unlike an HTML editor, such as HoTMetal, that works with a fixed tag set, XMetaL is meant to be used with any DTD and therefore requires customization. You will need a cascading style sheet and in most cases a set of macros for data entry for each new DTD. XMetaL supports the Windows Scripting Host, which means that you can write scripts in JScript, VBScript, Perl or Python to process XML documents or to create custom data entry interfaces.

XMetaL is intended to be integrated as a component of a broader XML solution, such as a content management system. The new Version 1.2 adds a built-in XSLT transformation engine.

## 1.6 Open Source Tools

In addition to the commercial offerings from vendors in the XML tools marketplace, there is a considerable body of other tools that have been produced to support SGML and XML content management by the academic and open source communities. Some examples of these include:

- **Bitflux Editor** - A browser-based WYSIWYG XML editor written in JavaScript and uses XML, XSLT, and CSS for rendering. It is usable with any XML document and features tables, lists, images, special chars, clipboard, undo/redo, and easy customization.
- **Ektron eWebEditPro+XML** - A browser-based XML word processor-like editor that enables business users to apply XML to Web content. It provides a user layer between the XML tags themselves and user actions. Scripting and commands work together to control which tags the user has access to, and where the tags can be used. Business users will not realize they are working with XML tags, but instead think they are working within a set of content parameters, definitions, and/or rules. Customization is required to implement the DTD and produce valid XML, but once this is done, there is little need for further integration.
- **GenDoc**(formerly GenDiapo) - An XML editor based on an existing project, MerlotXML. It can use two kinds of plug-ins (DTD and/or action). The DTD plug-in can be used to customize the editor for a DTD, and an action plug-in can be used to publish documents in HTML or PDF format. The editor is composed of three views: tree view, attribute view for current element, and a "styled view". The aim of styled view is to show the document with a visual aspect.
- **Morphon XML-Editor** - A validating WYSIWYG XML editor that lets you create and modify XML documents in an intuitive manner. Using DTDs and CSS, the editor guarantees the integrity of your XML documents and presents them in a consistent and user-friendly way. The XML editor is bundled with the Morphon CSS

Editor that can be used to customize your CSS, allowing you to change every aspect of the way the XML editor presents your document while editing. The CSS editor can also be used stand-alone to directly create CSS for the Web.

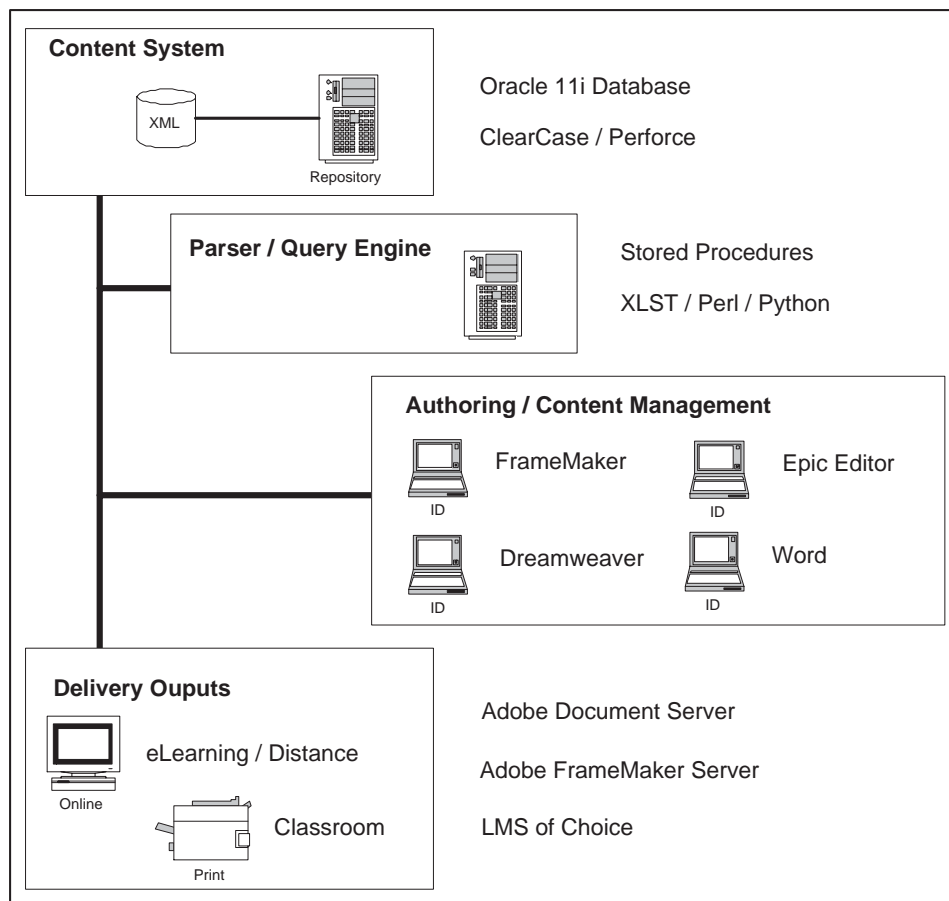
- **exchanger** - The eXchaNGeR XML browser is a browser and editor framework, written in Java, that visualizes elements in a XML document. The user can browse through and manage the visible elements in the document with external services, or she can make changes to the content of the XML document with the built-in XML editor.
- **Arsdigita CMS** - A powerful content management system. It has a task list for production staff to track assignments and the status of current work items; a site map browser to view and organize pages and content items and determine access control to branches of the site; a standard interface for creating, editing, approving, and deploying content items; a template manager for creating, editing, and organizing presentation templates and related assets; a metadata manager for viewing and defining content types and associations; a category browser for managing a hierarchy of subject headings that may be applied to content items; and administrative and management tools for creating and editing user attributes and tracking global work flow statistics.
- **OpenLMS** - An LMS made at the Department of Geography, NTNU. The system is a fully functional LMS with support for group collaboration, file sharing, distribution of lectures, and other supporting features. It is a good tool for distributing lecture notes to groups of students, and for facilitating collaboration for groups of students and teachers.
- **Moodle** - An LMS for producing Internet-based course Web sites. It is written in PHP and is easy to install and use on Linux, Windows, and Mac OS X. It has been designed to support modern pedagogies based on social constructionist theory, and includes activity modules such as forums, resources, journals, quizzes, surveys, choices, and assignments. It has been translated into 30 languages, with more on the way. Moodle offers a free alternative to commercial software such as WebCT and Blackboard, and is being used by a growing number of universities, schools, and independent teachers for distance education or to supplement face-to-face teaching.
- And many more<sup>10</sup>...

---

<sup>10</sup>See: <http://freshmeat.net> search topic Learning Management

## 2 Implementing a Unified Content Strategy

The XML Content Reuse System is composed of four parts:



Each piece of this larger system is associated with specific benefits and costs. There are basically two ways of achieving such a system: build your own from available components or buy one that does most of what you want and then customize that. If your organization has many specialized requirements and diverse processes, and your organization has considerable expertise and experience developing, implementing and maintaining software solutions, you will probably not save any money by buying a proprietary solution and then customizing it. If, on the other hand, your organization has much more general requirements for training, fewer audiences and simpler outputs, and your organization lacks in-house technical expertise and you normally contract out such projects, purchasing an off-the-shelf system may be a better solution.

## 2.1 Build Your Own

In order to devise your own content re-use system, you need to have some specific areas of expertise available:

- **Database Architect** - This individual creates a data library that exactly matches your DTD. These data tables are optimized to perform the most common search routines. The engineer should be experienced with hardware and network configurations appropriate to your organization's needs.
- **DBA** - The DBA is going to organize your query engine and make sure that all the routines operate properly to input and output data to your authoring and delivery environments.
- **Configuration Engineer** - This person configures and maintains the version control repository. This should be an expert in the software you have selected (ClearCase, Perforce, etc.,...). Many DBA's think they can do this job, but very few can. Configuration engineering is very important to making the whole system reliable and expandable.
- **Template Designer** - You will need one of these for FrameMaker and another one for Dreamweaver, if you use these products. Many organizations contract this task, an acceptable alternative, as many excellent consultants exist in this field.
- **LMS/Server Engineer** - This is an expertise that is generally provided (for a fee) by the software vendor that supplies the LMS or server platform. As noted before, Adobe has a wide range of services in supporting and training for their enterprise server products.

The principle advantage to be gained by undergoing the entire development process within your own organization is that you benefit most from the result of your analysis. You also build a core competency in developing and delivering learning objects.

The principle requirement for success for such a venture is buy-in from top management. There must be a commitment and a requirement to achieve a workable system in a modest time frame for a realizable cost. Successfully completing such a system results in the biggest gains in productivity and largest reduction in cost per training hour. Any large organization that has a sincere commitment to providing quality training programs, especially one that aims to increase the percentage of eLearning in its training offerings, should consider creating its own system.

Some of the main advantages and risks of developing your own content reuse system:

| Advantage  | Risk   |
|--|--|
| What you design is what you get. It is not necessary to wage an endless battle with a vendor over features or functionality.   | You are not purchasing a proven solution. Although the technology is sound, your implementation may fail.  |
| The system that results will be more extensible and flexible. As the needs of your organization grow and change, your system will accommodate these changes better.                          | Unless you exercise restraint, your system may outgrow your needs and become a monster that consumes more resources than it returns.                               |
| Your system is entirely within your control. Because you own all the source, you are not at the mercy of a third party.  | Your organization needs to be able to provide the development infrastructure to produce a satisfactory system and then maintain it enterprise-wide for many years. |
| Once the system is in place and in use, it is less expensive to maintain (unless you change it)  | You can budget expenses better with an outside contract than with an internal development project.   |
| Because your system is driven entirely by your own needs, you don't need to put up with evolutionary changes created for someone else's benefit, but with you must train your people to use. | When resources are scarce, you may find your development efforts are cut back precisely when you need more support.  |

## 2.2 Buy Existing System

The principal advantage to be gained by purchasing a system off-the-shelf is that someone else claims that it will work for you and further guarantees that they will support your implementation of their software. As with any vendor, you are negotiating a relationship of mutual benefit. Always spend more time researching the company and their references than you spend listening to the sales pitch.

Things to avoid when shopping for a system:

- Being the first customer, or being that vendor's first "big" customer.
- Buying a solution you do not understand - or one that the vendor is unwilling or unable to explain so that you can understand it.
- Becoming a client of a company whose primary imperative in software design is to lock you into their proprietary framework. This can be very dangerous, especially if the company disappears in 10 years.
- Purchasing a product that does not do some of the main things you are buying it to do, on promises that the company will customize it to do exactly what you want. At this point you might as well make it yourself.

Some of the main advantages and risks of buying an off-the-shelf content reuse system:

| Advantage   | Risk  |
|---|---|
| You are buying a proven product: it worked somewhere else.  | If it doesn't work for you, what's wrong with you?  |
| Your business processes are constrained to follow a proven model.   | Your processes are constrained whether or not that makes any sense for your organization.   |
| Without spending a large amount of your own capital, you benefit from receiving regular software updates.                               | The updates may wander further and further from your core needs, requiring more and more expensive customization.                         |
| You can budget a more or less fixed cost for support and custom services  | That budget may be inadequate to meet your organization's needs. The vendor may have no additional resources to meet extraordinary needs. |
| You are investing in a limited system, providing benefit against cost. Unlike a home grown system, which must be continually justified. | You cannot, with just a little more expense or effort reap any more result from the system.   |



## References

- [1] Adobe. Adobe solutions for document generation. [http://www.adobe.com/products/server/pdfs/document\\_generation\\_wp.pdf](http://www.adobe.com/products/server/pdfs/document_generation_wp.pdf), December 2003.
- [2] Adobe. *Safeco*. <http://www.adobe.com/financial/pdfs/safeco.pdf>, March, 2003.
- [3] Willaim Horton and Katherine Horton. *E-Learning Tools and Technologies*. Wiley, Indianapolis, IN, 2003.

# Index

abstract, 1  
adobe, document server, 5  
adobe, framemaker, 3  
adobe, framemaker server, 4  
adobe, golive, 5  
adobe, indesign, 5  
arbortext, epic editor, 7  
arsdigita CMS, 10  
  
bitflux editor, 9  
  
corel, xmetal, 8  
  
ektroneWebeditorpro+XML, 9  
exchanger, 10  
  
gendoc, 9  
  
macromedia, authorware, 7  
macromedia, dreamweaver, 6  
macromedia, flash, 7  
microsoft, other programs, 3  
microsoft, word, 2  
moodle, 10  
morphon, 10  
  
open lms, 10  
  
tools, 2  
tools, open source, 9  
  
unified content strategy, 11  
  
xml, home grown, 12  
xml, off-the-shelf, 13