

# THE **e**LEARNING DEVELOPERS' JOURNAL

Strategies and Techniques for Designers,  
Developers, and Managers of eLearning

JOURNAL™

## THIS WEEK — DESIGN STRATEGIES

### XML and Content Reuse Systems for Instructional Design

#### Part II: Implementing Content Repositories & Selecting Tools

BY HENRY MEYERDING

This is the second in a series of three articles for experienced instructional designers about XML and content reuse systems. The previous article in this series discussed the basics of XML as it applies to learning content management. XML, a derivative language from SGML that represents a simplified and optimized approach to creating databases, is an object language that allows you to create learning objects and transform them into a

variety of different forms. This article focuses on many of the methods and benefits of organizing content into repositories, and the tools that can be used to create the content stored there.

#### Taxonomy

Our scientific understanding of any topic is founded upon taxonomic processes: we take things apart to see how they work. We can gain a better understanding of the intricate parts of a whole system by examining its parts and then combining them together, gradually coming to understand how those parts interrelate.

In a very basic sense, what a content reuse system does is to scientifically divide content into associative, functional, or structural taxons. (See the sidebar on page 14, Definitions.) This taxonomy of information makes useful reuse feasible. The application of this useful taxonomy to enterprise information is what determines whether the content reuse system produces benefits for the organization or becomes just another expensive good idea.

All learning objects are defined by taxonomies. These taxonomies express the

*Continued on next page*

*Two weeks ago, the first article in this series introduced the basics and the terminology of XML for designing and using learning objects. In designing content reuse systems for e-Learning, the biggest issues are the taxonomy, the process, and the tools. Read this article for a complete review of these issues and a summary of tools that you may consider for your own content reuse implementation.*

**Publisher** David Holcombe

**Editorial Director** Heidi Fisk

**Editor** Bill Brandon

**Copy Editor** Charles Holcombe

**Design Director** Nancy Marland

**The eLearning Guild™ Advisory Board**

Ruth Clark, Conrad Gottfredson, John Hartnett,  
Bill Horton, Kevin Moore, Eric Parks, Brenda Pfaus  
Marc Rosenberg, Allison Rossett

Copyright 2004. **The eLearning Developers' Journal™**.  
Compilation copyright by The eLearning Guild 2004. All  
rights reserved. Please contact **The eLearning Guild** for  
reprint permission.

**The eLearning Developers' Journal** is published weekly  
by **The eLearning Guild**, 525 College Avenue, Suite  
215, Santa Rosa, CA 95404. Phone: 707.566.8990.

**The eLearning Guild** is an operating unit of Focuszone  
Media, Inc., 1030 Beatrice Street, Eagan, MN 55121.

**The Journal** is included as part of **Guild** membership.  
To join the **Guild** go to [www.eLearningGuild.com](http://www.eLearningGuild.com).

**The eLearning Developers' Journal™** is designed  
to serve the industry as a catalyst for innovation and as a vehicle for the dissemination of  
new and practical strategies and techniques for  
e-Learning designers, developers and managers. The **Journal** is not intended to be the  
definitive authority. Rather, it is intended  
to be a medium through which e-Learning practitioners can share their knowledge, expertise  
and experience with others for the general  
betterment of the industry.

As in any profession, there are many different  
perspectives about the best strategies,  
techniques and tools one can employ to accomplish a specific objective. This **Journal** will share  
these different perspectives and does not position any one as "the right way," but rather we  
position each article as "one of the right ways" for accomplishing a goal. We assume that  
readers will evaluate the merits of each article and use the ideas they contain in a manner  
appropriate for their specific situation. We encourage discussion and debate about articles  
and provide an Online Discussion board for each article.

The articles contained in the **Journal** are all  
written by people who are actively engaged in  
this profession at one level or another — not  
by paid journalists or writers. Submissions are  
always welcome at any time, as are suggestions  
for articles and future topics. To learn more  
about how to submit articles and/or ideas,  
please refer to the directions in the box  
on page 5 or visit [www.eLearningGuild.com](http://www.eLearningGuild.com).

way in which each object is understood,  
used and maintained. In evaluating how  
to construct learning object models for  
an XML repository, it is very important to  
understand that these models are used  
to define queries. The value of the system  
depends upon the ease and accuracy of  
queries. Many organizations discovered  
too late that they had expended  
substantial resources in creating an XML  
(or SGML) repository that provided no  
additional benefit over cutting and pasting  
documents from a file server because  
their content authors could not find  
anything in the repository.

### Repository design

A content repository has several different  
purposes:

- To store controlled versions of documents
- To store current versions of learning objects
- To store in-work versions of learning objects
- To publish content to Web servers
- To publish content to other servers (Learning Management Systems, abbreviated as LMS)
- To function as an ISO Repository

The most important reason for having  
a repository is to facilitate collaboration  
between content creators, editors and  
production staff. One mistake often  
made with a complex repository is to  
make customized views that are not  
shared between different team members.  
This can be frustrating and time-consuming.

Once everyone has gone through the  
arduous task of chunking and labeling  
their legacy content, this content needs  
to be put into a repository where it can  
be easily accessed. The best way to do  
this, for instructional designers, is to put  
the content into a version control system  
that is linked to a database.

ClearCase, for example, is a version  
control application that can present several  
different views of the repository for  
different uses. One view presents a virtual  
file server that contains all the most  
recent versions of the training documents.  
Another view presents selected documents  
to a Web server or LMS. Yet another view  
presents the XML database elements. Other  
views can be developed for specific uses,  
such as creating archives of content,  
presenting catalogs

of approved artwork or source content  
for other servers such as Adobe Document  
Server or FrameMaker Server.

The road to XML content reuse is simply  
a progression of responses faced by  
learning organizations. Generally speaking,  
there are six steps taken on the path from  
no content sharing and reuse to a  
comprehensive XML repository system:

1. File Server — A "shared drive" accessible to all team members with read and write permissions to all.
2. Version Control System — A collection of documents, stored by document version to protect against accidentally overwriting files.
3. Document Manager — A software system that provides different levels of access to documents based upon selectable criteria.
4. Learning Management System (LMS) — A system that provides access to learning content for students, authors, and editors. The modern LMS usually provides some kind of virtual campus paradigm.
5. Learning Content Management System (LCMS) — A system that divides up learning content into manageable components, which can be dynamically revised in some or all of its instances in the curriculum.
6. XML Repository — A system that applies content taxonomies to organize content into associative and structural classifications so that content can be created and managed with maximum efficiency.

Not every organization progresses through  
each step in an orderly manner. It is often  
the case that different groups within a  
learning organization implement different  
steps at different times, and then face  
significant challenges integrating the  
results. Table 1 on page 3 summarizes  
some of the objectives and limitations of  
each step in the progression.

When computer networks became  
common in the workplace, people abandoned  
the file cabinet for the file server. They  
soon learned that file servers have their  
own defects when it comes to sharing  
important information. The next logical  
step was to try to remove the most glaring  
defects of the file server by implementing  
a version control system. The version  
control systems made it safer to put your  
documents onto the network and easier  
to find things, but when large

numbers of people put large numbers of documents into the system, it became harder again.

Then came the document management system which made it simpler to find things, but which usually locked you into tools and processes that rapidly became outmoded. A good example of this last hurdle to progress was a large legal firm that implemented a complex macro-language-driven documentation system that interoperated with their document management system. When the next version of MS-Word arrived, they were very upset to find that there was no backward compatibility — so they remained with the older version of MS-Word for ten years.

Learning management systems (LMSs) are primarily student-facing applications. Their purpose is to present training to a student population and to track student performance. Over time, more and more content management facets have been sneaking into these learning delivery platforms. That is not their core function, which is to deliver existing content to students efficiently. Although learning content management systems (LCMSs) are designed to efficiently manage content, they suffer from a lack of flexibility and timeliness.

Everything that is true of document management tools locking you into particular tools and processes is also true of LMS/LCMS deployments, only much more so. Most LCMSs have their own content creation tools, which may be very well intentioned, but which also may fall very short of the functionality and finesse represented by other commercial applications. Of course, most will work with major content generators (more or less), such as MS-Word and Adobe FrameMaker, but they increase the complication of version upgrades by several orders of magnitude. This is a significant expense that must be factored into the cost of ownership and operation of these systems.

The best of the available LCMS systems are blended XML solutions. These systems use XML/XSLT technology as a transformation mechanism, but retain a proprietary data architecture for database functions. In this way, they have many of the advantages of XML technology, such as interoperability, SCORM-compliance, and access to XML enhance-

ments, and they can also customize the database engine to provide better system performance for content management functions. OutStart Evolution®, Aspen®, and learn eXact® are all examples of blended XML systems.

Once you have an XML repository, your repository can inter-operate with other systems, such as LMSs or even LCMSs, but the content is organized for your exclusive needs and convenience. If your needs or tools change, so can the repository. You have created for yourself an “Open Source” solution. For that reason, the XML repository is simpler and less difficult to upgrade than many proprietary solutions.

XML and SGML were developed specifically to provide a structure and methodology for content reuse. Many of the lessons learned from early SGML implementations were built into XML, which provides a more streamlined and less labor-

intensive means of achieving high quality content reuse.

Question: If XML repositories are so great, then why doesn't anyone market an XML repository as an LCMS?

Answer: Practically all LCMS vendors are organized according to a service consulting business model. They invest massive amounts of time and money to create efficient systems, which they practically give away for free. They do this so that they can sell you customizations, service, training, maintenance, and support. A pure XML repository system could be serviced and maintained by a wide variety of vendors, so they might never earn back the investment they made in creating the solution.

The proprietary product offer does tie the business to the vendor, but it also ties the vendor to the business because the vendor has a huge stake in the outcome of the LCMS implementation.

**TABLE 1** Objectives and Limitations in the steps to a XML repository system

Step	Objectives	Limitations
File Server	To permit access and sharing of files between many users.	Slow, insecure, and does not scale well.
Version Control System	To maintain different versions of the same document so that the newest (best) can be identified.	Complex to maintain and difficult to use when additional features are added.
Document Manager	To automate more complex features (and rules).	Proprietary — software does not keep pace with new tools and processes.
Learning Management System (LMS)	To improve the efficiency of training content delivery and progress tracking.	Can limit designers in terms of format or delivery methods, may not accommodate editing and version control well.
Learning Content Management System (LCMS)	To improve the efficiency of training development through content management and reuse.	Often includes a poor user interface; extensive customization required.
XML Repository	To provide content reuse, multiple output formats, and extensibility to react to changing needs.	Requires rethinking of the development model by designers.

There are some pure XML repository LCMS solutions that have been developed by the Open Source community (principally by and for academic institutions). They are more like do-it-yourself kits than a fully-developed product offering and do not offer the reliability, features, or performance of commercial off-the-shelf (COTS) solutions.

## Reusing content

Legacy content comes in many different forms. Most of these forms represent document instances. Most organizations attempt to maintain a repository of these document instances according to some meaningful hierarchy. ISO documentation standards are an example of this kind of document-centric hierarchy. If documents are correctly named, stored and updated, then the information they contain can be reused, but the process is slow, laborious and susceptible to human error. The utility of simple file sharing is inversely proportional to the number of documents to be shared.

When existing content is chunked, it usually begins in documents that are broken down into component topics and

then broken again into smaller pieces identified as introduction, main body, and transitions. Content should sound natural and appear to have been written specifically for each use. Content also is chunked by audience and complexity so that relevant material and more complex discussion can be added or removed easily.

Audience plays a big role in content reuse. Identifying specific blocks of information as appropriate or inappropriate for different audiences can simplify document creation immensely. It also is the hardest classification to accomplish.

For example, consider an Offer Brief: a document that quickly informs sales staff of new offers, pricing and conditions that apply to selling a product or service within a given market. These things are constantly changing. It is a major task to keep this kind of training content accurate and timely. Most of the documents have a similar look and feel. There may be specific types for different audiences or products, but a single item of information may find its way into 30 or 40 different presentations. Along the way it may get a different style — it may

appear in a table here and in a paragraph of text there, but the data behind it is identical. It is possible with each new iteration to do a keyword search through a documentation set and locate all known matches, then copy in the revised information. That usually takes too much time and trouble to be worth doing on a regular basis, unless it is very special information.

In comparison, with a properly constituted XML repository, the process is much more direct. Instead of working backwards from finished documents to find the appearance of specific content in context, the source content is already organized according to what it contains. The author goes to that container, revises it, refreshes the repository and the next time the document instance is called, it collects its source content from the updated source, applies the proper formatting, and compiles the finished document. All 30 or 40 documents that touch this same source content are thus automatically updated.

There was more work done in the very beginning to properly analyze and attribute the content, but as the content is

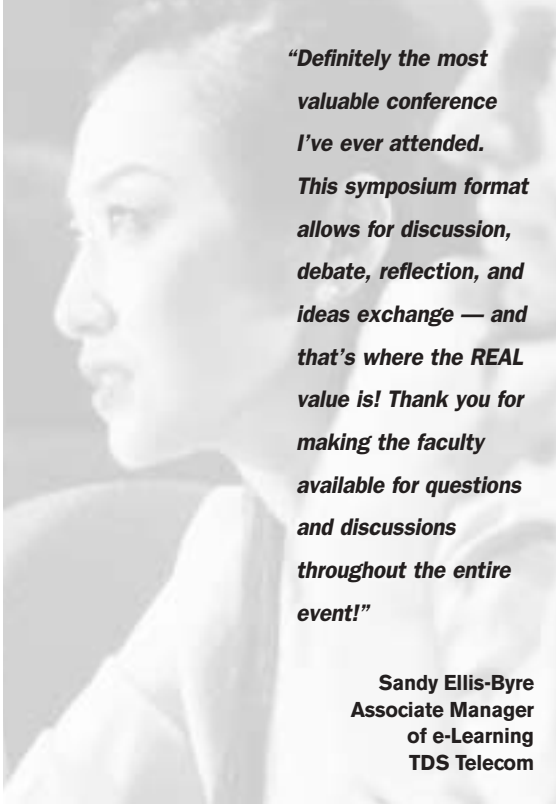
April 19 - 21, 2004 • Boston, Massachusetts

## THE E LEARNING FLASH DEVELOPER'S SYMPOSIUM™

Flash is one of the most-used web-tools for e-Learning development, but until now there hasn't been an event for you to really learn how you can leverage the power of Flash in your e-Learning applications. *The eLearning Flash Developer's Symposium* will get you "under the hood" of Flash to teach you the tips and tricks that will help you create more effective e-Learning. You will discover time-saving techniques, learn from practical case studies, and gain insights from e-Learning Flash experts. A wide variety of examples will be used to illustrate the many ways you can use this versatile development tool.

### Register Today!

To learn more about *The eLearning Management Symposium* go to  
[www.eLearningGuild.com](http://www.eLearningGuild.com)



***"Definitely the most valuable conference I've ever attended. This symposium format allows for discussion, debate, reflection, and ideas exchange — and that's where the REAL value is! Thank you for making the faculty available for questions and discussions throughout the entire event!"***

Sandy Ellis-Byre  
Associate Manager  
of e-Learning  
TDS Telecom

used to create more and more instance documents, those documents become progressively less expensive to create, manage and update. It makes it possible to do the previously unthinkable:

- Provide an individualized training syllabus for every employee.
- Implement weekly updates across training syllabi.
- Create monthly updates to training.
- Ensure global identification of misinformation.
- Provide personalized Web-based training tied to employee reviews.

By increasing the efficiency with which content can be created, the quality and timeliness of all the training deliverables can be increased without raising the cost into the stratosphere.

### Process

As H. L. Mencken said, "For every human problem, there is a neat, simple solution; and it is always wrong."

This section describes the development process used to implement the XML content reuse system. Each description includes a discussion of the costs and benefits associated with each process.

### Manual reuse systems

In traditional, project-oriented design settings, each new project was a separate entity. Analysis, development, and production were defined by the time line and requirements of each discrete project, and instructional designers produced design and content as an artisan custom-crafting a product for a customer. When this process has worked correctly, it has worked very well. Students receive curriculum that is specifically fashioned to address their needs. Trainers and designers can be student advocates at many different levels. Everybody wins. However, there are some important limitations to this methodology.

It is important to understand that these limitations and disadvantages are not a function of the skills or artistry of the designer. However dedicated and talented a designer might be, if armed only with a typewriter and a mimeograph machine, he will be at a disadvantage compared with someone, of perhaps more pedestrian talents, who is provided with computers and Web-based delivery options.

## Do you have an interesting strategy or technique to share?

*Get It Published in...*

# THE **e**LEARNING DEVELOPERS' JOURNAL™

This publication is by the people, for the people.

That means it's written by YOU the readers and members of *The eLearning Guild!*

We encourage you to submit articles for publication in the **Journal**.

Even if you have not been published before, we encourage you to submit a query if you have a great idea, technique, case study or practice to share with your peers in the e-Learning community. If your topic idea for an article is selected by the editors, you will be asked to submit a complete article on that topic. Don't worry if you have limited experience writing for publication. Our team of editors will work with you to polish your article and get it ready for publication in the **Journal**.

By sharing your expertise with the readers of the **Journal**, you not only add to the collective knowledge of the e-Learning community, you also gain the recognition of your peers in the industry and your organization.

### How to Submit a Query

If you have an idea for an article, send a plain-text email to our editor, Bill Brandon, at [bbrandon@eLearningGuild.com](mailto:bbrandon@eLearningGuild.com), with the following information in the body of the email:

- **A draft of the first paragraph**, written to grab the reader's attention and identify the problem or issue that will be addressed.
- **A short outline of your main points** addressing the problem or resolving the issue. This could be another paragraph or it could be a bulleted list.
- **One paragraph on your background or current position** that makes you the one to tell this story.
- **A working title** for the article.
- **Your contact information:** name, job title, company, phone, email. This information is to be for the writer of the article. We are unable to accept queries from agents, public relations firms, or other third parties.

All of this information should fit on one page. If the topic fits our editorial plan, Bill will contact you to schedule the manuscript deadline and the publication date, and to work out any other details.

**Refer to [www.eLearningGuild.com](http://www.eLearningGuild.com) for more details.**

At the same time, it must be admitted that the best tools will not make a poor designer produce excellent training content. Really good tools have been used to camouflage poor design. It is certainly easier for an incompetent instructional designer to produce much poorer training deliverables with an XML content reuse system than when working alone with MS-Word.

Assuming competent designers, some of the most important limitations and disadvantages of the cottage industry approach to instructional development are:

- **Inconsistency** — Since every project is independent of every other, it is very difficult to create and enforce standards. Even if templates are used, designers tend to create exceptions.
- **Inefficiency** — There are many opportunities for reusing content that are missed, either because designers are unaware of legacy content that could be adapted, or because the legacy content is in a format that makes it difficult to adapt to their current project.
- **Inaccuracy** — Because each project recasts some of the same information in a different way, there is no way to globally update information and reissue training when changes occur.
- **Scalability** — As workloads increase and staffing levels decline, there is no way to maintain output and quality levels. Designers become frustrated when they're unable to meet the ex-

pectations of their audience.

- **Tool Costs** — Reliance on outmoded tools, different versions of standard tools, and fringe tools complicates things, and makes people less efficient. The cost of maintaining learning materials sourced in multiple tools is enormous. Standardization on a few tools and methods makes a substantial difference to the production cycle.

### XML automated systems

Figure 1, below, describes a content authoring and delivery system for both online and hard-copy training deliverables. In this example, light blue indicates tools from Adobe, orange indicates tools from Macromedia, yellow indicates tools from Microsoft, and purple indicates tools for open source components or outputs. This is only one of many equivalent solutions.

The structured approach to instructional design is seen to have the following benefits, as William and Katherine Horton point out in *e-Learning Tools and Technologies*:

- The same courses are delivered across multiple media and delivery environments. Just because it happened to be developed by X using Y, this doesn't stand in the way of it being reused in a completely different environment or with different tools.
- The structured development model supports a consistent instructional design and development process.

Designers have many new options that come from an efficient production design.

- XML content can be analyzed and repurposed much more efficiently than legacy content. The content does not hide in a forest of words. When needed, new and legacy content can be efficiently blended to create educational tools to suit different needs of different student audiences.
- Learning content is organized for use. Related content is accessible. Related procedures and policies are obvious — as are conflicts and inconsistencies.
- Because the relationships between concepts and ideas are mapped according to the taxonomy by which the content was chunked, identifying content for reuse and the updating of legacy materials is significantly streamlined.
- Content conforms to Information Technology standards to ensure portability and long-term use.

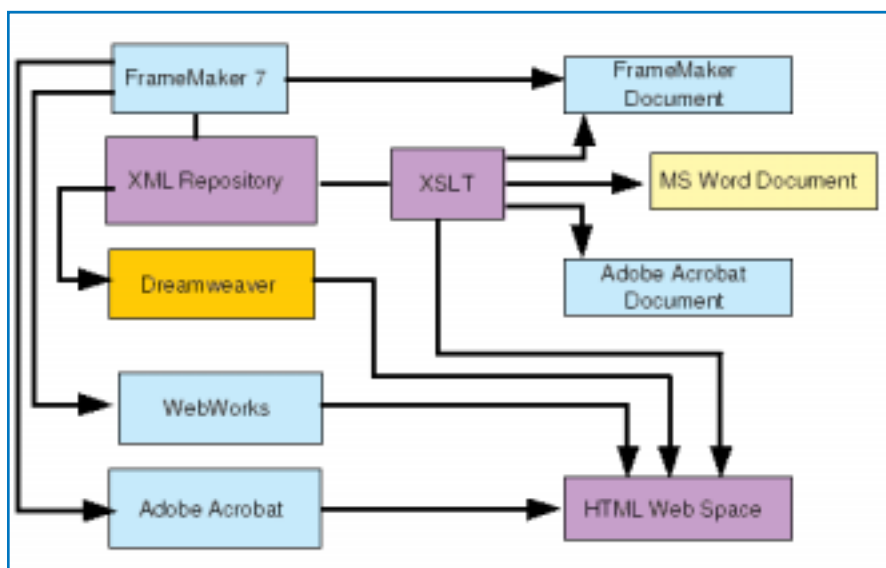
There are three steps in the process of implementing an XML content reuse system: 1) Analysis, 2) Chunking, 3) Operation. The process is very simple, in theory:

- A document type definition (DTD) is selected and tested.
- The repository is created using tables that mirror the DTD.
- Legacy content is converted to XML.
- XML content is placed in the repository.
- Users query the database to construct new documents.
- Users add new content to the repository as needed.

As mentioned before, the initial analysis is perhaps the most difficult stage of the implementation, and it is the one stage that has the most persistent effects. Having once decided upon the one and only way of parsing the content, staff members are carefully trained in how to accomplish the chunking of legacy content into the system.

### Legacy content chunking

Whether this chunking process is slow and manual or quick and automated really depends on how much legacy content was created by properly using standardized styles and templates. If practically none of the content was created using standard styles and templates, then



**FIGURE 1** This content authoring and delivery system produces both online and hard copy training deliverables.

there is a great deal of manual evaluation that must be done.

The most important aspect of the chunking process is to have the people doing the chunking UNDERSTAND what they are doing. This is best accomplished by providing them with thorough training, support, and supervision. Consistency is the key. Select a single process, train everyone in that process and execute the process without exception.

NOTE: The importance of thorough and consistent content editing increases by several orders of magnitude when content is entered into the database. Enter it wrong once... use it wrong many times. Michael Hughes says it best:

“Organizations that implement highly configurable or customizable products need to rely on their software vendors to meet the early training needs of the planners and technicians. To the degree that they wish to own or control product configuration, customization, and the ongoing support of those modifications, they also need to be prepared to invest in the staff development required to enable those capabilities.”

There are two approaches to legacy content that are usually successful:

- Identify a small select team of designers who specialize in converting content. They do nothing else until the original body of required content has been put into the database.
- Spread the conversion duties among all the design team. Each member converts documents among their other duties, but at least a fixed minimum number of hours per week.

The advantage of the first method is that you generally obtain a more consistent conversion with fewer errors. The advantage of the second method is that you train your entire group in the XML database and process. You also may learn some things early on that allow you to modify the database or your processes so that they are more applicable to your training.

As with any complex operation, when there are advantages, there are also risks. The risk inherent in the first method is that it may result in a fully functional content base but with no one trained to use it properly. The second method risks creating a database with so many inconsistencies that it is practically useless. The correct method for

each organization depends upon the technical background of the team and their workload. Organizations with lower levels of technical proficiency and higher per capita workload generally do better with the first method.

### Using chunked content

The theory of developing new documents from legacy components is fairly simple, if the repository is implemented properly. First, the designer needs to know what previous training this new training is similar to. This is accomplished by querying the database and seeing what existing content comes fairly close to the current need. If it is completely new and dissimilar from other training, then the designer gets nothing from the repository but templates. Having made a shrewd guess about some other similar training, the designer has to define how this new training is different from the similar training that has been identified.

One method of handling the query process is by a Web page containing drop down field list properties. Define five or six of these properties and then add in some more specific customizing terms, click SUBMIT and get back a list of matching content. It is just like doing a Web search, except that the Web you are searching is a discrete database. What is returned from the search can take many different forms: FrameMaker

documents, raw XML, Word documents or HTML. When the query results in more “hits” than desired, then reformulate it to be more specific. If little or nothing results, then try a more general query until you get the desired results.

The authoring process is iterative, a succession of repetitive operations performed to collect, modify and upload new content. (See Figure 2, below.)

As time goes by, and the authors and production people get used to using the system to produce the required results, productivity increases and frustration decreases. There will be some people who simply cannot adjust to the new work methods, just as there were some very talented people who could produce marvelous typed documents but who could never quite make a word processor work right.

Some authoring environments, such as Epic Editor, work from the data structure to the content. At the beginning, these tools can be difficult for some designers to understand and use efficiently. After the designers become familiar with the database structure, they rapidly learn to navigate through the maze of information they encounter on cross-functional teams to find the parcels they want. In practice, authors working with common, standardized documents rapidly learn the five or six elements they must identify to generate the greater portion of their training. It is more difficult,

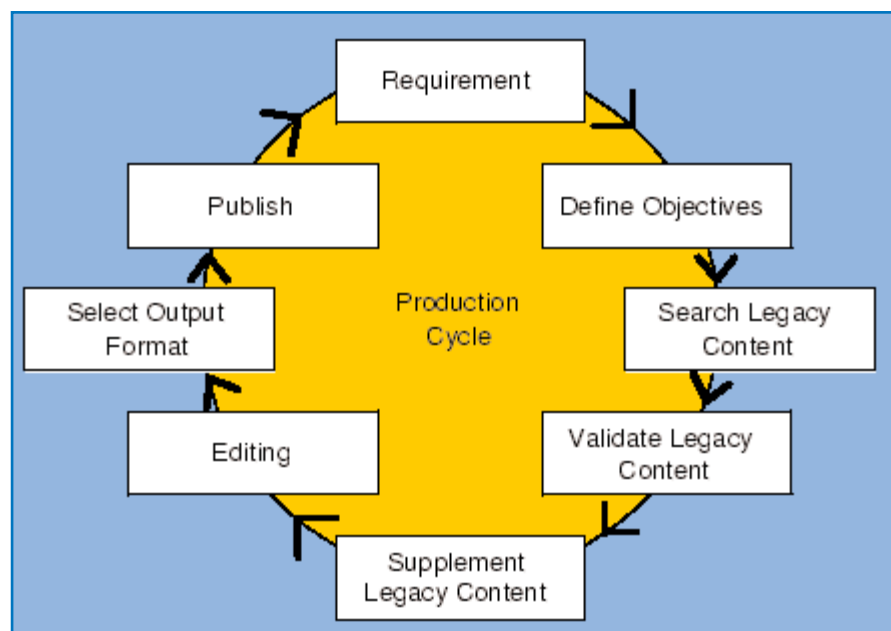


FIGURE 2 The authoring process is an iterative cycle.

in the beginning, than cutting and pasting content, but once you get into your stride, it becomes 10 times faster and easier to do your job. Even in a pure XML environment, designers still find the ability to easily query the database invaluable.

It should be noted here that no content management system can stand in for the designer's knowledge and understanding of the corpus for which training is developed. XML has no real impact upon the analysis or discovery phases of new training development. XML is only a set of tools. Having the skills to manipulate those tools does not in and of itself result in training, any more than reading a manual makes you an expert.

How the content is organized into new instances is a question of authoring tools, not XML.

### Multi-sourcing

Multi-sourcing has been the Holy Grail of the documentation industry for a generation. Simply put: information goes in

the hopper, press button A and a good marketing document results. Press button B and you get a User Manual, and pressing button C creates the getting-started pamphlet that goes in the box with the product. The heart of multi-sourcing is content reuse.

Until the advent of SGML, content reuse was impractical — until the advent of XML, content reuse was out of the reach of all but the largest organizations and institutions. But in the last 30 years, tremendous advances have been made in content reuse technology to enable multi-sourcing of documents.

Whenever the notion of content reuse is raised, one hears the same kinds of objections voiced time and time again. These kinds of questions are entirely typical and a natural reaction to the concept of content management and reuse. Adopting an XML or other reuse system asks people who already know how to do something well to change their process and to adopt methods they do not know.

The following Frequently Asked

Questions (FAQs) are the four most common questions asked by seasoned designers:

- How will XML help me to tailor my materials to meet the needs of my audience?

When you are creating training now, if you have a good, useful piece of content that speaks to the same point in another class — don't you copy and paste it in? If you could do this more often, and maintain the same quality of output, would you do it? We all do that, within the body of our own work, and sometimes from other authors, too. We use our own documents as a source for reuse because we are intimately familiar with them. We know we can find that great paragraph we used to describe that weird thingamajig. More seasoned authors annotate their own works with notes that help them find those good opportunities to reuse content. If we work for long enough with another designer who does the same, we can get so that we can read each other's notes and make

## e-Learning for e-Learning Professionals...

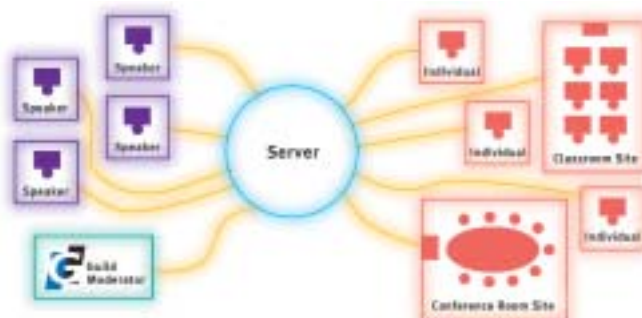
# The eLearning Guild's ONLINE FORUM™ SERIES

The eLearning Guild has created *The Guild Online Forum Series*, a new series of online events that will be held throughout 2004. On the 2nd Thursday of every month (except January) you can register to participate as an individual, or as a group, in a one-day "virtual conference" that includes four highly interactive seventy-five minute sessions designed to explore a specific topic.

### Here's how the Online Forums work:

#### Individual or Site Registration:

*Participate as an individual or you can pay a site fee, set up your meeting room, and have your e-Learning team participate in an Online Forum as a group!*



*Here's a brief description of the first Online Forum in the series...*

FEBRUARY 12, 2004

### Managing and Delivering High-Impact Synchronous e-Learning

Acquire a comprehensive understanding of how to manage and deliver synchronous e-Learning. Discover how instructional design for the synchronous environment differs from design for the classroom or asynchronous e-Learning. Learn how to develop the skills needed to be an effective online trainer.

**Target Audience:** This Online Forum is geared for anyone who is exploring the acquisition and deployment of synchronous e-Learning technologies, and for those looking for ways to maximize their current use of synchronous technologies.

*To learn more about each upcoming Online Forum and to register, go to:*

[www.eLearningGuild.com](http://www.eLearningGuild.com)



more use of each others' work as reuse content. XML, in this sense, is like a common form of notes with which we annotate each others' work so that we can access it and reuse it, when that is the best thing to do.

- How can I leave out technical information in one document, but include it in another?

This becomes something of a tool question. What are you authoring with? You may collect the entire content for a document and run through it, individualizing it for this instance, cleaning up any transitions and outputting the result. Instead of copying and pasting text between documents, you are attaching document objects to one another like a jigsaw puzzle. Pieces that work well consecutively have the right "shape" to fit together that way. They add up together, you edit them as needed and produce the finished product.

- What prevents this from resulting in documents that seem mechanized and impersonal?

This is not a machine imitation of human communication. This is human beings using a system of shortcuts to make their work easier and more productive. Certainly it can sound mechanized and artificial, but it doesn't have to be so. When the same item or process is described identically in five different places in four different classes — is that mechanical or is it using repetition to reinforce?

- What about the shifting voices of the authors; won't that cause confusion?

Do you have more than one instructional designer on your staff now? Are your students confused by having to attend classes created by different people in isolation? When everyone else's work is more available and when opportunities for collaboration and knowledge sharing are facilitated by the system, instead of being hampered by it, will that not help these different people find a more common voice. If you never sing in a choir, you never get the knack of singing like everyone else.

### Single-sourcing

All printed documents since Gutenberg were multiple copies of a single original source (single-sourced documents) until quite late in the 20th century. Advanced

---

**...it must be admitted that the best tools will not make a poor designer produce excellent training content. Really good tools have been used to camouflage poor design. It is certainly easier for an incompetent instructional designer to produce much poorer training deliverables with an XML content reuse system than when working alone with MS-Word.**

---

printing technology allowed compositors to create multiple versions of documents by reusing the same printing plate sources in different permutations. Computers made practically anything possible, but only a comparatively tiny slice of the possible became routine.

In an enterprise environment, there are many uses to which information is put. Some of those uses include documentation, training, knowledge-base applications and marketing. Traditionally, these disparate uses have all maintained their separate knowledge management environments. As a result, the information provided by these various sources is usually inconsistent — and in the worst case it results in considerable misinformation.

No one would think of using typewriters and mimeographs for corporate communication anymore, although these were once ubiquitous. In the near future, single-source systems will seem just as antiquated.

When these information sources are unified into a single repository from which all outputs derive, significant improvements in efficiency, consistency and overall quality of information result.

Also, when the costs of implementing the content repository are spread among different organizations within the enterprise, a greater return on investment naturally occurs.

Communication is the unstated core competency of every successful business. When the information about its products, processes, policies and procedures is available to all associates, this has a unifying effect on all organizations within the enterprise. Although the process and deliverables of different organizations vary tremendously, their need for accurate and timely information is identical.

In its best form, the XML content repository can be a significant competitive advantage to an enterprise, particularly one that operates in diverse markets. In this sense, the economies and productivity conferred to the training organization are a byproduct of a larger benefit to the entire enterprise.

### Tools

Our view of the possible is shaped by our tools. — *Carl Sagan*

This section analyzes some of the common tools that can be used with the XML content reuse repository. There are many tools available and one size does not fit all. The choice of tools is an important one, because the tools will have the biggest and most immediate effect on the designers. For that reason, it is very important to include designers in the tool selection process.

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 agonizing revelations — and at least three ways to do something:

1. The RIGHT way — the way originally envisioned by the developer and facilitated by the program. This way works best in the long run.

2. The WRONG way — the way that someone found to make it work, because they didn't know what the right way was. This way complicates editing and later revision of the material.

3. The OTHER way — the way that outwits the program and allows you to do something that should not be done, but

needs doing. This way has everything wrong in common with the WRONG way, with the added disadvantage that it may actually make your application or their documents unstable.

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 pragmatic 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.

## **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 .rtf format 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.

People quite often learn to use Word by trial and 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

---

**A structured document view for creating valid XML, several different levels of styles, and the ability to discard exceptions to styles are three of the most important features that impact XML. For this reason, Adobe FrameMaker is the WYSIWYG authoring tool of choice for XML applications.**

---

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.

Because users seldom use Microsoft Word properly, it is rarely 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 invalidate 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.

Even though it is the most popular word-processing application on the planet, many people consider that Microsoft

Word has no place in an enterprise XML content reuse system. In this view, using word processor technology to author content objects is counter-intuitive, inefficient, and ineffective. Regardless, people resist trading tools, even when they have good reason to do so. Some dedicated XML editors, such as Epic (see below) even include filters to import Word content to XML. Indeed, there has been a significant amount of effort to create robust, reliable conversion tools for making XML extracts from Word documents. The newest generation of blended XML LCMSs, such as OutStart Evolution, includes an impressive amount of bi-directional filtering of content to and from MS-Word.

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.

## **Adobe and FrameMaker**

A structured document view for creating valid XML, several different levels of styles, and the ability to discard exceptions to styles are three of the most important features that impact XML. For this reason, 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.

A couple of points about valid XML documents may be important to the reader's understanding. 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 DTD. All valid content is well-formed. Not all well-formed content is valid.

Adobe FrameMaker imports the XML data elements into a template. That template defines styles associated with the element definitions in the element defini-

tion document (EDD). (The EDD is the imported copy of the DTD used by FrameMaker to validate the XML content.) This means that it is not 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. This only makes sense when you are using FrameMaker as your publication tool. Otherwise, what the author sees is only the best approximation the XML programmer can make with an XSLT of the normal output from FrameMaker styles.

Adobe FrameMaker can export files directly through Webworks to HTML. (See the notation under Dreamweaver — later in this article — about working with Webworks-generated source code as an HTML output.) This is an option for training projects that rely extensively on inter-related print and online media. It can be easier to coordinate and publish the required learning materials 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 (such as bookmarks, different kinds of linking, different security modes, and so forth) 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 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 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 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 a top-level commitment to pursue an Adobe-enabled XML solution, you'll find that the kind of support and expertise available from Adobe is unequaled elsewhere in the industry.

**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. This allows designers to access current information in a printable form, which 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.

FrameMaker server works with the FrameMaker software on the desktop to provide more groupware solutions to enterprise publication challenges. It is designed for working in a distributed net-

working environment, and it provides convenient document management functions from within FrameMaker itself that make many group collaborations simpler to manage.

**Adobe Document Server.** Adobe Document Server 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, including graphics and audio to produce bi-fi multimedia 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 into a single document instance that the user receives.

Adobe has many products and services designed for XML-based solution environments. XML technology and Adobe software work together in a highly complementary fashion (See Figure 3, below, representing the Adobe Document Server solution implementation of XML). This is not an accident.

**Adobe InDesign and GoLive.** Page-oriented software, used for brochures and presentations, allows a lot of flexibility and precision in placing content on each page. Document-oriented software, used for manuals and other books, is intended for larger, more complex documents

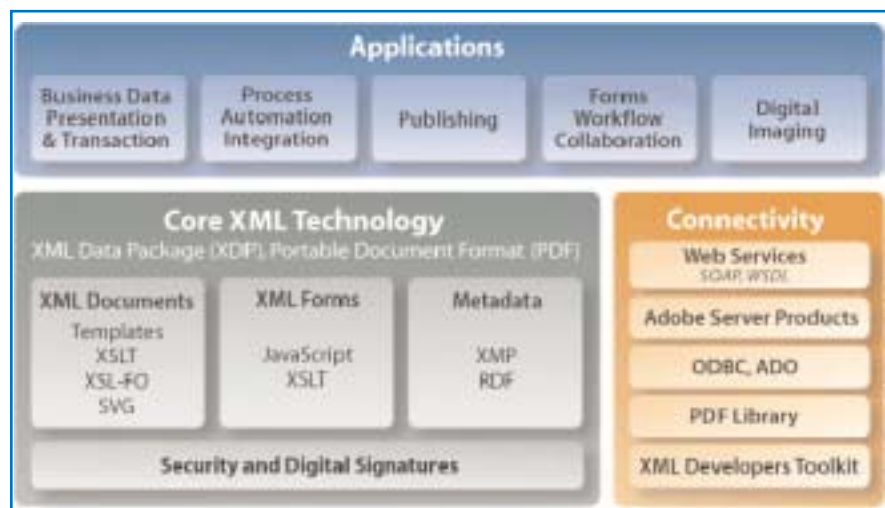


FIGURE 3 Adobe Document Server solution implementation of XML

where management of cross-references, indices and other features is more important.

Adobe InDesign is page-oriented software that includes 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 Scalable Vector Graphics (SVG) and sharing of native Photoshop and Illustrator files and it 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.

Because it is a page-oriented development tool, as opposed to document-oriented, Adobe InDesign is a good choice for small (two- to five-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. Both GoLive and Dreamweaver will send you scurrying into the source code at edit time. The WYSIWYG editing mode is very nice and handy, but it is maddeningly imprecise. If lo-fi 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.

**Macromedia Dreamweaver.** For online content, Macromedia Dreamweaver is one of the most popular WYSIWYG HTML editors. Unfortunately, like Microsoft Word, it is often misused. 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, some-

---

**The big advantage of 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.**

---

one may have extensive experience creating Web content with Dreamweaver but not have a clue about using templates.

Dreamweaver's WYSIWYG editor is, as noted above, imprecise and you cannot make many edits without having recourse to the source code. 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, not whether it is valid XML.

HTML generated by Webworks from Adobe FrameMaker files may not behave well in Macromedia Dreamweaver, as noted earlier in the section "Adobe and FrameMaker." Dreamweaver has less tolerance of HTML code that it interprets as badly-formed XML. (HTML is well-formed when it conforms to the syntax of the version of HTML supplied in the document definition.) 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

e-Learning. The combination of Webworks and FrameMaker works best for document-based learning, where a large volume of information must be provided to the student as reference material.

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.

**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.

**Arbortext Epic Editor.** Unlike other text editors that have been stretched to fit the function of authoring XML content, Arbortext Epic Editor was designed from the ground up as an 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 user friendly, 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 authoring 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 that 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 and 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 or PDF, and Web or 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 process requires human intervention and exercise of good judgment. However, 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

### Summary of tools

Table 2, right, provides a quick reference for some of the important tools that have been discussed in this section. The following abbreviations are used in Table 2:

- Word — Microsoft Word, current XP version, some features available at additional cost.
- FM — Adobe FrameMaker + FM Server + Adobe Document Manager
- ID/GL — Adobe InDesign/GoLive combination
- DW — Macromedia Dreamweaver
- AT — Arbortext Editor
- XM — XMetaL
- OS — Open Source tools, in aggregate

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, allowing users to see a visual representation of their output. Given that the designer understands XML and the learning content, Epic can out-perform 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.

**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 document 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 JavaScript, 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.

### 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.

NOTE: This section is more technically

**TABLE 2** Summary of tools (refer to abbreviations, left of table)

Advantage	Word	FM	ID/GL	DW	AT	XM	OS
Documentation	✓	✓	✓	✓	✓		
Format conversion	✓	✓	✓	✓		✓	
LMS integration		✓	✓				✓
Native XML		✓		✓	✓	✓	✓
Online Help	✓	✓	✓	✓	✓		
Support available		✓	✓	✓			
Training available	✓	✓	✓	✓		✓	
Valid XML support		✓			✓	✓	✓
Well-formed XML		✓		✓	✓	✓	✓
WYSIWYG	✓	✓	✓	✓			
XML training		✓					
Disadvantage	Word	FM	ID/GL	DW	AT	XM	OS
Costly		✓					
Extensive prep required		✓				✓	✓
Conversions required	✓		✓	✓			
Inefficient	✓		✓	✓			
Not an XML application	✓		✓			✓	✓
Steep learning curve		✓			✓	✓	✓
Training required		✓				✓	✓
Uncertain future						✓	✓
Very technical					✓		✓
Primary output	Word	FM	ID/GL	DW	AT	XM	OS
Paper	✓	✓					
Web		✓	✓	✓			
Other online			✓	✓		✓	✓
XML					✓	✓	✓


oriented than the preceding sections. In general, open source solutions require a more technically oriented user. To make up for this, they may contain extremely powerful features that are not available in other products at any price, let alone for free. Many Open Source software solutions are available without acquisition cost, though the real cost of ownership may be considerably higher.

Some examples of Open Source tools include:

- Bitflux Editor — A browser-based WYSIWYG XML editor written in JavaScript that uses XML, XSLT, and CSS for ren-

dering. It is usable with any XML document and features tables, lists, images, special characters, 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, 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 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 — A LMS made at the Department of Geography, Norwegian University of Science and Technology (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 — A 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 — see <http://freshmeat.net> and search topic Learning Management.

(Editor’s note: The third and final article in this series will be published in **The Journal** on February 16, 2004.) 

## SIDEBAR *Definitions*

**Taxon** — A taxon is a category of information. An internally consistent collection of taxons constitutes a taxonomy.

**Open Source solution** — An open source application is one in which you have access to every line of code. If you have the expertise, you can modify it in whatever way is necessary for your own purposes, rather than bartering with a vendor to get changes implemented second-hand.

**COTS** — Commercial Off-The-Shelf

**ISO** — International Standards Organization. This is a network of national standards institutes from 147 countries working in partnership with international organizations, governments, industry, business, and consumer representatives. It is a bridge between public and private sectors.

**Bi-fi** — This term refers to selectable low-fidelity/thin bandwidth or high-fidelity/wide bandwidth distributed content.

**Lo-fi and hi-fi Web content** — Lo-fi Web content is primarily text with a few graphics and moderate interaction; it is suitable for thin client delivery. Hi-fi Web content is highly graphical with strong user-interaction; video, Flash, and Authorware content are hi-fi.

**WebDAV** — Web-based Distributed Authoring and Versioning

# About the Guild

## REFERENCES

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.

Adobe. *Safeco*. <http://www.adobe.com/financial/pdfs/safeco.pdf>, March, 2003.

Daly, Paul and Watchorn, Helen. "Word and XML: Making the 'twain meet.'" *Journal of XML Europe 2001*, 2001.

Horton, William and Horton, Katherine. *E-Learning Tools and Technologies*. Wiley, Indianapolis, IN, 2003.

Hughes, Michael. "Keeping just-in-time from being way-too-little." *Performance Improvement*, 42(6):37-40, July 2003.

ISO Online. "International Organization for Standardization." ISO Online Web Site, 2003.

Paille, Gerry; Norman, Solvig Prescott; Klassen, John Maxwell. "The Effect of Using Structured Documents (SGML) in Instructional Design." <http://naweb.unb.ca/proceedings/1999/paille/paille.html>, February, 1999.

## AUTHOR CONTACT



Henry Meyerding is an instructional designer, with experience in the telecommunications and software industries. Trained in Engineering, Henry's career gradually changed from doing engineering to explaining to others what engineering had accomplished. More than fifteen years in the field of technical documentation provided many lessons in information systems, most of them relevant to learning design. The focus of Henry's career came to center on instructional design largely through participation in rework of documentation masquerading as training, so that it turned into actual training. He found that he enjoyed working with students to overcome obstacles to learning and as a result Henry has concentrated on developing blended learning solutions. He lives in Sultan, WA with his wife and four children. After being outsourced for the second time to the same outsourcing firm, he is currently seeking other opportunities and working on a book project to explain practical instructional design theory and method to technical documentators. Contact Henry by email: [hwm@goomba.com](mailto:hwm@goomba.com).

**Additional information on the topics covered in this article is also listed in the Guild Resource Directory.**



**The eLearning Guild™** is a Community of Practice for designers, developers, and managers of e-Learning. Through this member-driven community, we provide high-quality learning opportunities, networking services, resources, and publications. Community members represent a diverse group of instructional designers, content developers, Web developers, project managers, contractors, consultants, and managers and directors of training and learning services — all of whom share a common interest in e-Learning design, development, and management.

## The eLearning Developers' Journal™

The Guild publishes the only online "e-Journal" in the e-Learning industry that is focused on delivering real world "how to make it happen in your organization" information. The Journal is published weekly and features articles written by both industry experts and members who work every day in environments just like yours. As an active member, you will have unlimited access to the Journal archive.

## People Connecting With People

The Guild provides a variety of online member networking tools including an Information Exchange and a Job Board. These services enable members to discuss topics of importance, to ask others to help them find information they need, and to provide leads to other members.

## Resources, Resources, Resources

The Guild hosts the e-Learning industry's most comprehensive resource knowledge database. Currently there are over 4,500 resources available. Members have access to all of these resources and they can also post resources at any time!

## Guild Research

The Guild has an ongoing industry research service that conducts surveys on a variety of topics each year. These topics are identified by the Research Committee. The data collected is available for all Members.

## It's About Leadership

The Guild draws leadership from an amazing Advisory Board made up of individuals who provide insight and guidance to help ensure that the Guild serves its constituency well. We are honored to have their active engagement and participation. The Guild has also established two committees made up of active members who help steer its events program and research efforts.

## Discounts, Discounts, Discounts

Guild Members receive discounts on all Guild conferences and on other selected products and services. Your Guild membership will save you 20% off the list price of Guild events!

**Become a member today! Join online at [www.eLearningGuild.com](http://www.eLearningGuild.com).**