

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 I Introduction to XML and Repository Technologies

BY HENRY MEYERDING

XML as a content reuse methodology is a very complex and technical topic. Instructional designers seeking to discover the impacts of XML technology on the field of instructional design are very often frustrated by the available technical white papers which usually fail to speak to the practical needs of instructional design. This three-part series is written for experienced instructional designers who wish to understand content reuse in an XML

environment. These articles describe XML and discuss how XML can be used in an instructional design setting to manage and facilitate the definition, use, and distribution of learning objects. This first article explains the basic underlying concepts at a non-technical level and relates those concepts to instructional design processes. The second article discusses taxonomies, processes and tools that can be used in conjunction with different source repositories. The final article describes different implementations and how to determine the return on investment for content reuse systems.

Introduction

"Write it once... use it many times."

This simple statement describes the function and attraction of extensible markup language (XML) as an authoring methodology.

In its simplest form, a learning object is discrete information that speaks to a specific learning objective. That being said, learning objects are not simple things, but are complex constructs of information, presentation and interaction. Designers traditionally have seen themselves as artisans who create unique

Continued on next page

Content reuse continues to be a hot topic, but the technology involved is new and complex. This is the first of a three-part series that explains how content reuse tools are used in instructional design, and looks at how to think about a unified content strategy to facilitate the process. Read this part in order to get the high-level overview of content reuse.

THE eLEARNING DEVELOPERS' JOURNAL

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.

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.

DESIGN | strategies

learning tools for each new learning situation. They have been very slow to make the transition in their thinking to a systematized approach to the development and delivery of learning. They have also spent long weary years learning how to use particular tools and will resist giving up their hard-earned virtuosity, even when the tools in question are obviously a barrier to meaningful improvement.

At the same time, the enterprises for which most instructional designers work have been under increasing pressure to provide training more efficiently. Training departments that are unwilling or unable to deliver substantial increases in efficiency risk being replaced by outside contracting firms that promise to deliver these efficiencies.

XML has become a standard means of information interchange within the computer industry. Using XML to create and manage learning objects is not just a theory, but has a long track record of use in the real world. It offers greater efficiency without reducing the quality of the training deliverables.

But working within an XML environment requires a change in perspective. Instead of approaching each task as the resolution to a specific obstacle to learning for a specific audience, the designer must analyze the task in a wider context. With whom does this audience share this learning requirement? How can this learning obstacle be resolved for all students, not just the students involved in the immediate situation? If this objective does not apply to other groups, are there components within it that do apply to a wider audience? What existing training can be pulled into this task and modified to work, without affecting the quality of the learning? What other training is relevant to the content currently under development? How can these new content objects be fitted to other uses?

Some experienced designers, who are more used to routine and repetitive iterations of vast waves of training materials, may find very little in the previous paragraph that speaks to their job description. Their organizations have identified deliverable requirements, methodologies and audiences for them and the designers are charged with producing the required quantity of training that meets a relatively low quality standard. The attractive feature of XML to these designers is

that it offers a way to respond to practically impossible demands for training with substantially less drudgery, thus allowing designers to build at a higher standard of quality, which is (or which should be) always on the nice-to-have list.

Instead of being a creative artist fashioning unique responses to specific learning requirements, the designer becomes a production professional who analyzes the learning needs of a specific group as those needs relate to the generalized requirements of the entire learning community. The importance of designers clearly understanding the capabilities and rationale behind the content reuse system cannot be stated strongly enough.

The changeover into an XML content development and production environment really represents the same quantum leap in capability as was achieved by replacing typewriters with word processing on computers. Managers must be evangelists of XML technology's liberating capabilities and they must be zealous in training their staffs to understand these capabilities. *There has never been a system so good that it could not be rendered totally ineffective by resistant participants!*

One of the incidental benefits of operating in an XML learning object environment is that designers are exposed to content created by other designers much more than in traditional project environments. Properly managed, the specialized understanding of different teams is more effectively shared, and the quality of the output is increased.

Terminology

Before launching into a discussion of XML in more depth, it is important to understand some of the terminology that will be used throughout this series of articles.

Attribute — The characteristic of an XML element that defines the content. Example: If the elements are class, type, and color; corresponding attributes might be toy, rubber ball, and red.

Chunking — The process by which legacy content is tagged for inclusion in the content database.

Content — Content is information. It may take the form of text, graphics, audio, or video.

Database — A hierarchical distribution

of data arranged in relationships that provide quick access to information of interest.

Document — Strictly speaking, when working in XML there is only one super-document that contains all the content. This content fits into a common structure. We extract pieces of this super-document and publish it as document instances, which may be either static or dynamic.

Dynamic Instance — When publishing a document instance, it may contain information that changes continuously. In defining the document publication instance, it may be desirable that every time a user opens the instance they see the most recently updated information. This contrasts with static instance.

Element — An XML element is a definition for content. Any piece of content may be defined by one or several elements. Examples: class, type, and color.

HTML — Hypertext Markup Language — developed from SGML as a means of conveying information on the Web.

Learning Object — A functional component of training curriculum; a building block. Each learning object generally

Managers must be evangelists of XML technology's liberating capabilities and they must be zealous in training their staffs to understand these capabilities. There has never been a system so good that it could not be rendered totally ineffective by resistant participants.

addresses a specific learning goal. Just how specific a goal varies from system to system.

Legacy Content — Content, usually in electronic form, such as text, graphics, audio or video that has been developed outside of an XML content environment.

Legacy content often resides in short-lived proprietary formats, which make reuse or conversion problematical.

Metalanguage — The language that is used to talk about (expressions of) another language, the object language. XML contains and identifies content, but the XML is not the content.

Parse — To divide into components from a larger set based upon some identifying feature or content.

SCORM — Sharable Courseware Object Reference Model — A set of specifications for developing, packaging and delivering high-quality education and training materials whenever and wherever they are needed.

Static Instance — When publishing a document instance, it may serve as a standard or reference. In defining the document publication instance, it may be desirable that users see a single, unchanging document, until any changes have been approved by a ruling/governing body. This contrasts with dynamic instance.

SGML — Standard Generalized Markup Language — the international standard metalanguage for text markup systems.

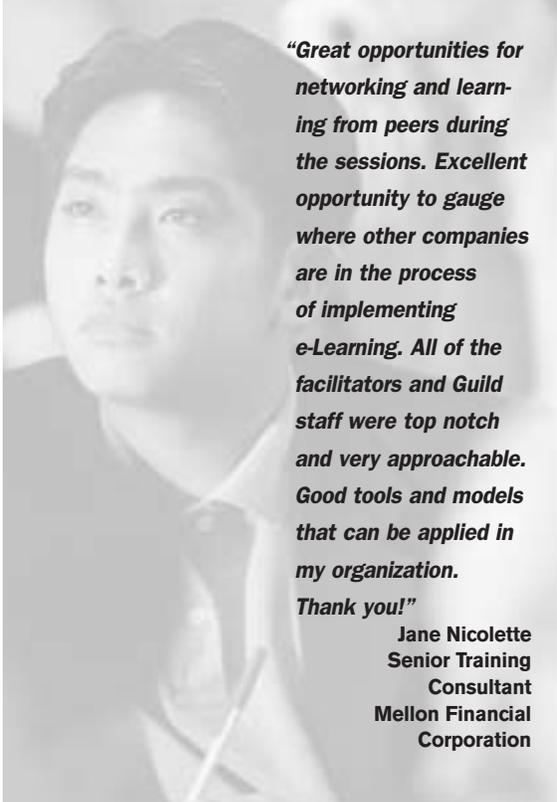
February 18 - 20, 2004 • San Diego, California

The **E**LEARNING MANAGEMENT Symposium™

The eLearning Management Symposium is focused on managing the strategies, people, and processes that will enable you to produce and deliver successful e-Learning in your organization. This symposium will provide you with a seamless learning environment that will enable you to drill into the hottest issues facing e-Learning managers today. This is going to be an intensive learning experience.

Register Today!

To learn more about *The eLearning Management Symposium* go to
www.eLearningGuild.com



“Great opportunities for networking and learning from peers during the sessions. Excellent opportunity to gauge where other companies are in the process of implementing e-Learning. All of the facilitators and Guild staff were top notch and very approachable. Good tools and models that can be applied in my organization.

Thank you!”

Jane Nicolette
Senior Training
Consultant
Mellon Financial
Corporation

Taxonomy — a system for naming and organizing things into groups that share similar qualities.

WAP — Wireless Application Protocol — the standard for accessing the internet with wireless devices, e.g. mobile phones. (Not to be confused with Wireless Access Point hardware.)

XML — eXtensible Markup Language — developed as a more manageable subset of SGML.

The next section introduces XML and discusses some of the features of XML that make it particularly appropriate for learning object development and learning content reuse.

XML basics

What is XML? The typical definition of eXtensible Markup Language (XML) says: “XML is a new World Wide Web Consortium (W3C4) specification. XML is a pared-down version of SGML, designed especially for Web documents. It enables developers to create their own customized tags to provide functionality not available through HTML.” (Harry Newton, *Newton’s Telecom Dictionary, 17th Edition.*)

This definition provides a lot of information about XML, but it doesn’t help us understand much about XML and its capabilities. XML was designed as a database metalanguage. It was designed as a means of structuring content so it

could be put into and be retrieved from a database in a form that was useful for content reuse. Information content can be text, graphics, audio, video, or complex constructs of all these learning components.

The principal difference between XML and HTML is that XML uses “smart tags.” Smart tags convey information about the content they contain. Because of this, you can use the structure you create to put your content into an easily retrievable form. One exciting aspect of XML is the ability to define your content your way, creating custom tags for different kinds of instructional objects such as objectives, test questions, feedback and other common training content components.

Your content management system reads the smart tags and parses your content into useful chunks that you can assemble into subsequent documents. When you need the same (or similar) content, you construct a query of your content database. You then review the resulting content and if it matches your current need, you use it. If you find nothing useful, you add new content for your current document — and for future use.

By using smart tags, it is also possible to define very specific criteria for making recursive changes. A branding change, which might take weeks to implement across an entire curriculum, can be implemented in minutes. Editorial and style changes can be very exactly implemented in the precise circumstances defined by the editors. Legal reviews can be conducted on exemplar text, which is then recursively edited throughout the content library.

Each of the content elements is consistently tagged so that it fits together with other elements to form consistent document instances. In other words, each content object contains an introduction, main matter, illustrations (if any), conclusion and transitions. Several content objects are aggregated to form a document instance. Given a different use, it may be necessary to slightly modify some of the introductory or transitional materials. You also can structure your learning objects so that they can be specifically relevant to different user groups or to audiences of different aptitudes. The next document instance you require may contain the same content

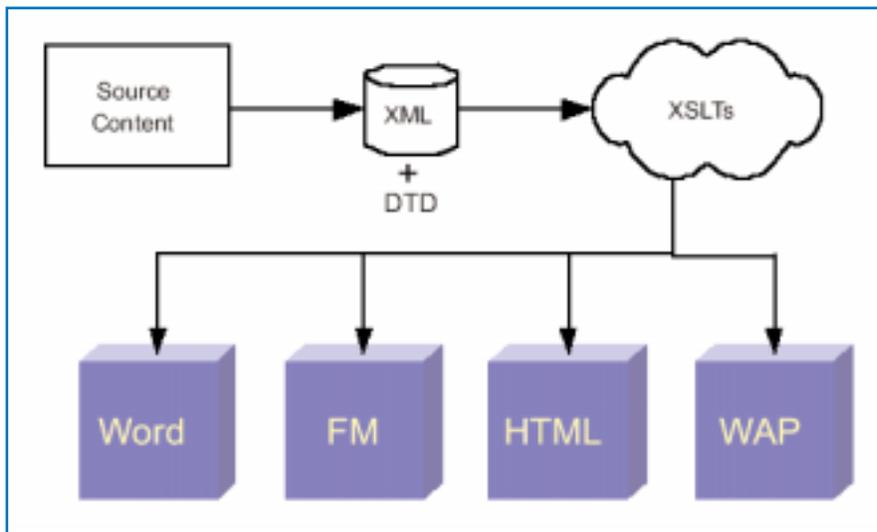


FIGURE 1 Source content is transformed into document instances or learning delivery instances through the XML object components. (FM = FrameMaker document, WAP = Wireless Application Protocol document).

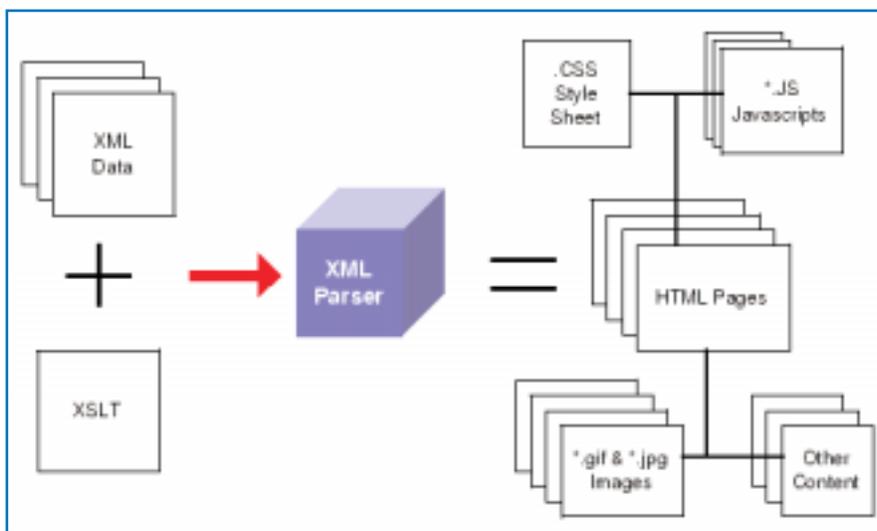


FIGURE 2 XML data and an XSLT are parsed to become a Web site.

objects, which have been modified, just a little here and there, to meet the requirements of the different audience.

The process of defining these structural components is analogous to creation of animation cells from layers. Each layer contains a different quality of information pertaining to the same object. These objects are then added together to produce an instance of learning delivery. (See Figure 1 on page 4.)

There are three main components of XML objects with which instructional designers need to be concerned: XML, DTD and XSLT.

The XML file contains the content, just the content and only the content.

The Document Type Definition (DTD) specifies how the XML is structured.

The XML Style sheet (XSLT) contains all the formatting information for the output document instance.

Document Type Definitions (DTD)

Most XML documents (and all SGML documents) refer to an external document type definition (DTD). The DTD provides a list of the elements, attributes, comments, notes, and entities contained in the document. It also indicates their relationship to one another within the document. In other words, a DTD is the grammar of an XML document. Without a DTD (or schema, see below), all you have is data. The DTD is rather like the header row in a table that identifies what lies below it. It does not itself contain any data, but instead makes the data that follows useful.

DTDs are complex and difficult to create from scratch. Luckily, there are thousands of open-source DTDs from which to choose. Most of them are related to a specific industry or enterprise. Unless you have unlimited time and capital, there probably is no reason to even contemplate making a custom DTD that models your core organizational elements. The most widely distributed DTD is DocBook, which originally was developed for documenting computer software. Many popular DTDs, such as the Telecom Interchange Markup (TIM) are derivatives of DocBook. Because XML is extensible, you can begin with a DTD that meets your core requirements and then make minor modifications and additions to it at a later date without invalidating all your XML.

Do you have an interesting story to tell?

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, 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 for more details.

The DTD defines the taxonomy that you will use to parse information about the subject matter into re-usable form.

XML Schema. Considerable confusion exists about XML Schema. Simply put, XML Schema represents another, more flexible, way to accomplish the same functions as DTDs. Just like the DTD, the XML Schema is an external document referenced from the XML document. In terms of learning objects, XML Schema do very little additional, useful things, as compared with DTDs.

Where XML Schema do come into their own are in validation of numerical data. For example, if you are passing instructions in XML format to a machine that cuts precision parts from blocks of steel, it would be handy to have that

data automatically validated before it is handed to the machine, since accidentally sending the wrong numbers can ruin not only the production piece but the very expensive machine as well.

It also is true that most of the document-handling software has been developed in an XML-with-DTD environment. If, in the future, as some forecast, XML Schema replace DTDs, there will be simple tools to convert DTDs to equivalently behaving XML Schema. For the time being, in regard to learning content management, it is not necessary to pay any attention to Schema.

XSLT

XSLTs are the method used to transform raw XML data into useful content

— creating a Word document or a FrameMaker document from the same source files is accomplished simply by applying two different XSLTs to the same XML data files. They are referred to as style sheets, but they are not style sheets in the same way as a cascading style sheet defines the look of an HTML document instance.

It is important to note that authoring in XML divorces the content from the format. The reason for this is quite simple: consistency. When you make a change to the source content of a series of deliverables, you should be able to produce those new versions with a minimum of intervention — ideally with no intervention. This is accomplished by creating an XSLT for each document instance type and then processing content through that common XSLT for all versions of that output deliverable type. The same revised content can be processed through XSLTs to produce an update for four different courses, which may be either Web-based or paper-based.

This is a departure from traditional word processing and publication tools, which merge these two operations into a single work flow. Some tools used to author XML, such as FrameMaker, allow the designer the illusion of authoring content with format, but all formatting is defined in the XSLT, which authors very seldom modify. Rather than spending endless hours fiddling with documents to make the styles come out right, developers work with the content to make that right. When the words and graphics are right, they are done.

XML is data and contains no formatting. However, because it is structured data, the formatting can be applied to it programmatically. The XML data and the XSLT are parsed to become a Web site. The Web site contains the HTML pages, images, javascripts, style sheets and other content associated with a full-featured Web site. (See Figure 2 on page 4.) The same content can be parsed with a different XSLT to result in fully featured WAP Web content for delivery to a portable device, such as a cell phone.

An XSLT also can convert the same content to a Microsoft Word document (or any other document format that is desired). (See Figure 3, left.)

Lastly, an XSLT can be used as an intelligent query to result in the XML con-

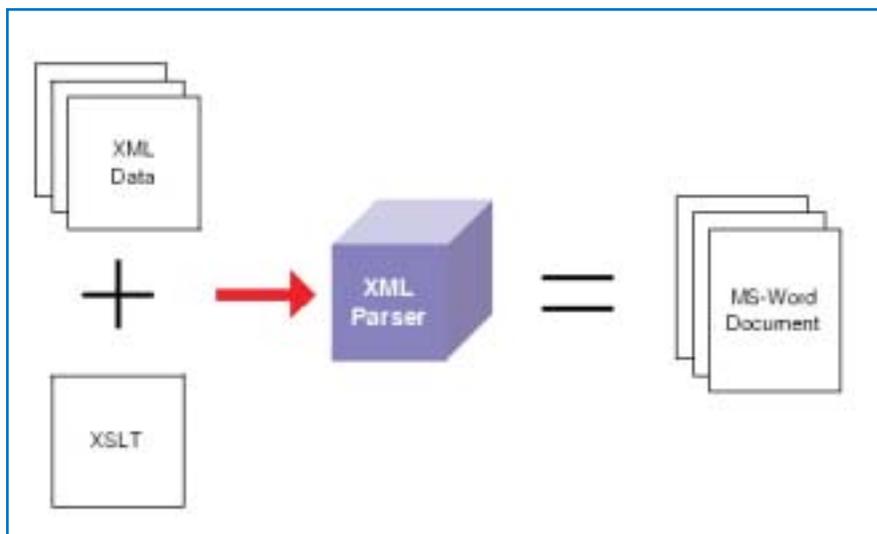


FIGURE 3 The XML data from Figure 2 and a different XSLT can be parsed to become a Microsoft Word document.

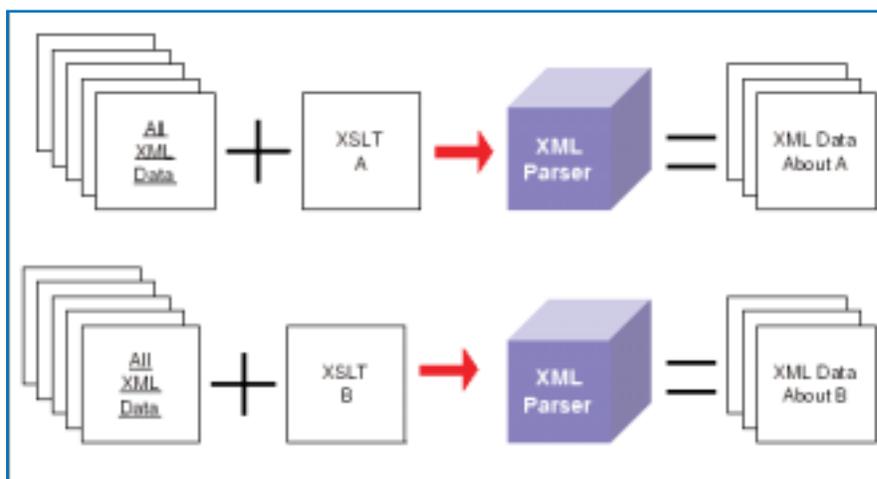


FIGURE 4 An XSLT can be used as an intelligent query to pull specific data from a database.

tent that conforms to a specific set of rules. These rules can be very complex and can be a more efficient way to query data than a direct Structured Query Language (SQL) query to the database. You may have all your content about a subject in XML format and use an XSLT to separate out from the total content only those passages which conform to specific criteria. (See Figure 4 on page 6.)

What about...?

The following sections describe several initiatives and protocols that are related to XML and that are of particular importance to e-Learning designers and developers.

SCORM. The Sharable Courseware Object Reference Model (SCORM) is very prominent in the marketing literature of a lot of content management vendors. To hear these vendors talk, if your XML is not SCORM compliant, there is something wrong with you. SCORM is a set of standards, implemented in XML, which

[The] designer becomes a production professional who analyzes the learning needs of a specific group as those needs relate to the generalized requirements of the entire learning community. The importance of designers understanding the capabilities and rationale behind the content reuse system cannot be stated strongly enough.

passes instructions to a Learning Management System (LMS). It is analogous to an application program interface (API).

APIs are shortcuts for doing complicated things. For example, programmer A writes an accounting program and wants to allow programmer B to write a program that queries the accounting program for specific kinds of information. Programmer A does not want programmer B to know exactly how his accounting system works, so he writes an API that allows any application to send the accounting system a very specifically worded question, to which the accounting system will reply with the desired information. Programmer B now only needs to know how to ask the question, and not how the accounting system handles the operation of answering it.

If your organization selects a SCORM-compliant LMS, then your repository should contain an XSLT to present your content with additional SCORM elements that enable the LMS to know how to deal

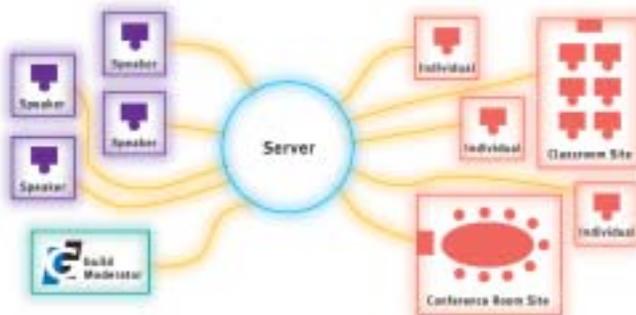
e-Learning for e-Learning Professionals...

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:

Site Registration:
Imagine! You can pay a single 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

with it. The nice part about SCORM is that it is an open standard. If you decide that your current LMS does not meet your needs, you will be able to select another SCORM-compliant LMS and only have to tweak your repository here and there to make it all work correctly.

SOAP. The Simple Object Access Protocol (SOAP) is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses.

Plain language version: You own an Internet-connected soft drink machine. Once a day, you want your soft drink machine to send information to the inventory management package. You then want the inventory management package to talk to your payroll system so that the salesman for the territory gets paid the right amount. Either you buy all these systems from the same vendor, or you buy each one from the lowest bidder and make sure they all talk SOAP to one another. You then only need one programmer to make it all work instead of a staff of fifteen programmers to write it from scratch.

Although SOAP is usually associated with Microsoft, it is actually maintained by a consortium of companies, of which Microsoft is one.

.NET. Microsoft's "dot net" (.NET) is a "software platform." It's a proprietary language-neutral environment for writing programs that easily and securely interoperate. Rather than targeting a particular hardware/OS combination, programs instead target ".NET," and run wherever .NET is implemented. In this way, it is just like Java. .NET is also the collective name given to various bits of software built upon the .NET platform. These are both products (Visual Studio.NET and Windows.NET Server, for instance) and services (such as Passport, HailStorm, and so on).

.NET uses XML in many instances, especially when complex data is transferred. It is a case of using what works best to accomplish some tasks. Other than this, it has very little to do with XML, despite the mounds of marketing that issue forth from Redmond which try to make .NET and XML seem synonymous.

SGML. Standard Generalized Markup Language (SGML) is the root language for many other languages, such as HTML and XML. It has been around since 1968 and is still a universal tool, both platform- and system-independent, for describing text. SGML is rather like Latin — the language from which other languages have been derived. Like Latin, it is more difficult to use in a contemporary context than are its children. (See Figure 5, below.) Also, there are many more tools being developed every day to do things in XML, which is no longer true of SGML.

XML owes much of its functionality and usefulness to the lessons learned by people using SGML.

Conclusion

This has been a fairly high-level overview of the underlying concepts of content reuse in an XML environment. The terminology, XML basics, and the information about the components, initia-

BIBLIOGRAPHY

1. Eckstein, Robert, with Casabianca, Michel. *XML Pocket Reference*. O'Reilly, Sebastopol, CA, 2001.
2. Horton, William and Horton, Katherine. *E-Learning Tools and Technologies*. Wiley, Indianapolis, IN, 2003.
3. Kampherbeek, Jan. *Crash Course in XML*. <http://www.spiderpro.com/bu/bu/xmlm001.html>, June, 2001.
4. Means, W. Scott and Harold, Elliotte Rusty. *XML in a Nutshell*. O'Reilly, Sebastopol, CA, 2002.
5. Muellner, Leonard and Walsh, Norman. *Docbook, the Definitive Guide*. O'Reilly, Sebastopol, CA, 1999.
6. Newton, Harry. *Newton's Telecom Dictionary, 17th edition*. CMP Books, New York, NY, 2001.
7. Piskurich, George M. *Rapid Instructional Design*. Jossey-Bass/Pfeiffer (Wiley), San Francisco, CA, 2000.
8. Ray, Erik T.. *Learning XML*. O'Reilly, Sebastopol, CA, 2001.
9. Reigeluth, Charles M.. *Instruction-design Theories and Models, Volume II*. Lawrence Erlbaum Associates, Mahwah, NJ, 1999.
10. Sperberg-McQueen, C. M. and Thompson, Henry. *XML Schema*. <http://www.w3.org/XML/Schema>, April, 2000.
11. Tidwell, Doug. *XSLT*. O'Reilly, Sebastopol, CA, 2001.
12. W3C. *Simple Object Access Protocol (SOAP) 1.1, W3C Note*. <http://www.w3.org/TR/SOAP/>, May, 2000.

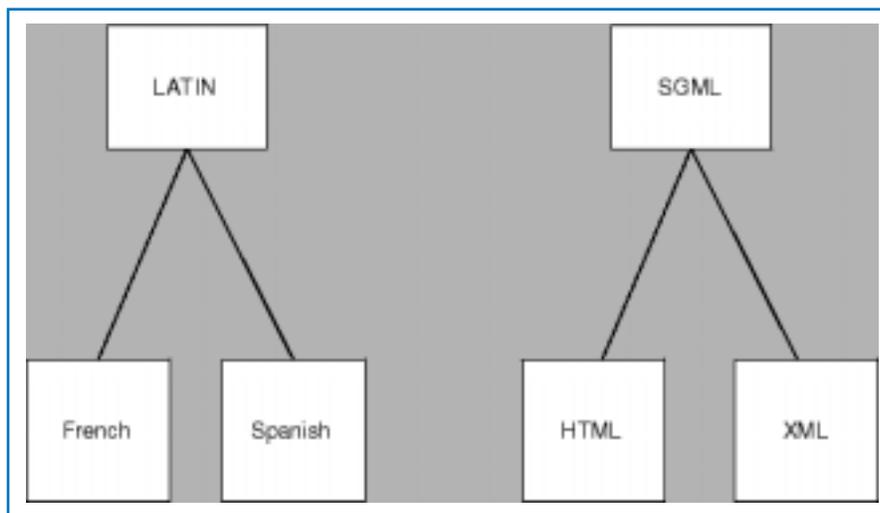


FIGURE 5 SGML is a parent to other markup languages.

About the Guild

tives, and protocols will be applied to instructional design processes in the next two articles in this series.

Part 2, to be published February 2, 2004, will address the way in which content reuse systems divide up content in order to provide benefits for the organization. It will also review a large number of proprietary and Open Source tools that support content reuse.

Part 3 will discuss the implementation of a unified content strategy to support instructional design. This will appear in the February 16, 2004 issue of **The eLearning Developers' Journal**. 

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 methodology to technical documentators. Contact Henry by email: 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.

Become a member today! Join online at www.eLearningGuild.com.

THANK YOU TO THESE GUILD ENTERPRISE SPONSORS



Because the most dramatic learning happens through first-hand experience, everything we do is focused on creating experiences — engaging, challenging and yes, fun, experiences — that people will take back to work and use every day to improve their performances.

**www.alleninteractions.com
Contact: Jackie McMillan
jmcmillan@alleni.com
800.204.2635**

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!

Join today at www.eLearningGuild.com!

To learn about **Guild** sponsorship opportunities, please contact David Holcombe at dh@eLearningGuild.com or call 707.566.8990.