

XML and Content Reuse Systems for Instructional Design Part III Creating a Unified Content Strategy

Henry Meyerding

January 2004

Abstract

This is the last of a series of three articles about XML and content reuse systems for experienced instructional designers. The first part explained the underlying concepts of XML and content reuse systems and related those concepts to the instructional design process. The second article discussed taxonomies, processes and tools that can be used in conjunction with different source repositories. This final article describes different implementations and the determination of the return on investment for content reuse systems.

Copyright 2004 - Henry Meyerding

All rights reserved - Neither this article nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the author.

1 Introduction



Write it once ... use it many times.

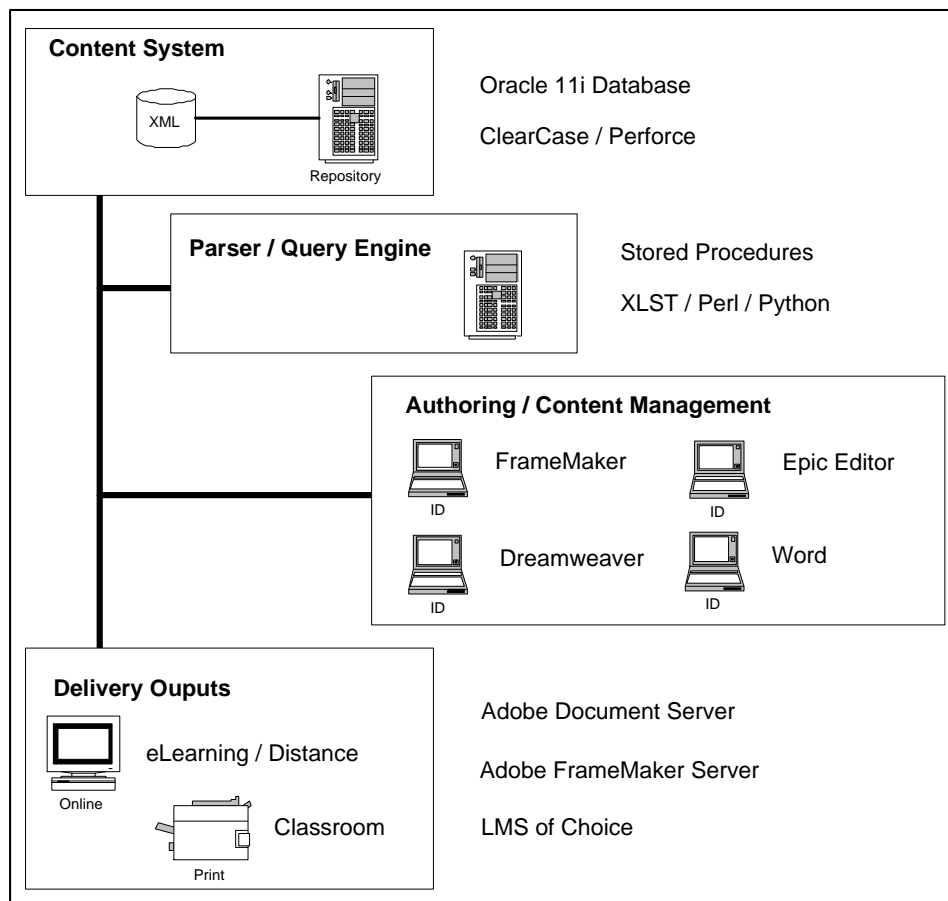
The previous two articles in this series discussed the basics of XML, taxonomies, processes and tools used for generating and modifying content. XML is the organizational methodology used to construct repositories of learning objects that have been created using a taxonomy.

The learning taxonomy is used to classify information according to a set of rules. Repositories for these derived learning objects represent the culmination of a gradual process that began with simple file sharing. The reason for entering into all this extra work is to allow the learning organization to be capable of responding more quickly and flexibly to the needs of students. Different tools can be used and none is perfect. The proper training and supervision in how the tools are used will impact the finished product much more than the tool election will.

This article concludes the series by examining how the content in the repository can be used as part of a Unified Content Strategy whose aim is to create measurable return on the system investment.

2 Implementing a Unified Content Strategy

The XML Content Reuse System is composed of four parts:



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

2.1 Build Your Own

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

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

You only get the full value of your analysis and planning if you carry out the results of that research by developing your own solution. Any other approach compromises your results. You also build a core competency in developing and delivering learning objects.

The principle requirement for success for such a venture is buy-in from top management. There must be a commitment and a requirement to achieve a workable system in a modest time frame for a realizable cost. Successfully completing such a system results in the biggest gains in productivity and largest reduction in cost per training hour. Any organization (of more than 10 training content designers) that has a sincere commitment to providing quality training programs, especially one that aims to increase the percentage of eLearning in its training offerings, should consider creating its own system.

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

Advantage	Risk
What you design is what you get. It is not necessary to wage an endless battle with a vendor over features or functionality.	You are not purchasing a proven solution. Although the technology is sound, your implementation may fail.
The system that results will be more extensible and flexible. As the needs of your organization grow and change, your system will accommodate these changes better.	Unless you exercise restraint, your system may outgrow your needs and become a monster that consumes more resources than it returns.
Your system is entirely within your control. Because you own all the source, you are not at the mercy of a third party.	Your organization needs to be able to provide the development infrastructure to produce a satisfactory system and then maintain it enterprise-wide for many years.
Once the system is in place and in use, it is less expensive to maintain (unless you change it).	You can budget expenses better with an outside contract than with an internal development project.
You build a great deal of specialized competency in your designers and production staff	Replacing that expertise can be very difficult to do.
Designers and developers work together, keeping one another current in skills and development within the XML world	Designers and developers spend a lot of time integrating new software version updates and other less-productive tasks.
Because your system is driven entirely by your own needs, you don't need to put up with evolutionary changes created for someone else's benefit, but with you must train your people to use it.	When resources are scarce, you may find your development efforts are cut back precisely when you need more support.

2.2 Buy Existing System

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

Things to avoid when shopping for a system:

- Being the first customer, or being that vendor's first "big" customer.
- Buying a solution you do not understand - or one that the vendor is unwilling or unable to explain so that you can understand it.
- Becoming a client of a company whose primary imperative in software design is to lock you into their proprietary framework. This can be very dangerous, especially if the company disappears in 5 years.

- Purchasing a product that does not do some of the main things you require it to do, on promises that the company will customize it to do exactly what you want. At this point you might as well make it yourself.

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

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

The deciding factor in whether to buy a vendor product, as opposed to creating a custom solution from other components, is resources. Getting any new system implemented is going to require resources. If the resources are not going to be available within your own organization, then you will need to purchase those hours from external vendors. Creating your own custom solution is going to require many more hours of development than implementing a vendor solution. If your training department is small, or your organization does not have the budget to spend on developing future capabilities, at the expense of deliverable training hours today then you may have insufficient resources to properly design and implement your own system.

No system that has insufficient development resources allocated to it can compete with an off-the-shelf product. In developing that solution, the vendor can amortize development costs across many different clients. Continuing development and maintenance costs are similarly shared. Many organizations have a cultural bias toward purchasing turn-key solutions, even if they do not perform as well as custom applications. Regardless of the technical benefits bestowed by one kind of system or another, it is often better to pick the right solution that matches the business realities of the enterprise.

2.2.1 Case Study

Business A is an aerospace company with a very large and capable IT organization. It has a history of developing very complex, very customized solutions that meet exacting business and regulatory requirements.

Business B is a large manufacturer of consumer products, with a relatively small and under-appreciated IT organization. It regularly purchases software solutions and maintenance contracts that provide for the special needs of specific user communities within the organization.

When Business A went into the content management marketplace, they did extensive research of many different vendors with competing products. They had a tendency to "study a product to death." The IT and engineering organizations generated thousands of pages of conflicting and contradictory requirements, which no vendor was able to meet.

When Business B went into the content management marketplace, their aim was to find a state-of-the-art product that they could purchase to perform a limited set of very specific tasks. They concentrated on vendors with associations to their existing vendors and very quickly narrowed the choice down to two competing products.

Business A purchased an off-the-shelf product whose vendor promised to customized to fit the needs of the enterprise. The IT organization fought the project tooth and nail from start to release. When eventually implemented, the system was largely ignored by many of the divisions of the organization, despite having been specifically tailored to meet their needs. The Director of information services and communication then used this software as a club to bring each of the disparate organizations into line - to streamline their procedures and to regularize their methods for producing documentation and training for each of their markets on 5 continents. Though training productivity initially suffered, after all was said and done, the system achieved a 40% increase in training hours per designer. The resulting training was consistent, won numerous industry awards and was instrumental in creating a truly global training organization.

Business B hired a team of three consultants to work with every division to develop a customized solution from open source components. In the process of analyzing the communication and training needs across the different divisions, the team discovered large pockets of inefficiency and waste. During the three year development cycle, the development program cost the organization approximately \$17 million. In the ensuing 5 years, it produced higher quality training deliverables consistently throughout the enterprise and contributed significantly to lowering the training costs for new employees by 38%, resulting in an average cost savings of \$12 million per annum. By selecting this solution path the company identified training as one of their core competencies.

3 Return on Investment



If we don't change direction soon, we'll end up where we're going. – Professor Irwin Corey.

In today's competitive training market, it is increasingly important to provide management with measures of performance that can be used to quantify the return on investment in training. Traditional resistance on the part of instructional design practitioners has resulted in lowered funding levels and the dilution of the importance once accorded to training programs. This result is diametrically opposed to the stated aims of the training development organization: to achieve the greatest performance improvement. Therefore, if the aims of the training development organization are to be realized, the demonstration of measurable return on investment for training is just as important as creation of processes whereby performance can be improved.

3.1 ROI Basics

Return on Investment (ROI) is a percentage calculated as the ratio of net program or project benefits to program or project costs times 100. What are the benefits associated with training? Generally, training is supposed to provide people with the capacity to perform a particular function. Theoretically, the better the training is, the better the function is performed. Performing a function better is usually measured by the time required to successfully execute tasks, the number of mistakes made per thousand operations performed, and the ability of the student to extend these skills into new and more complex operations.

Collection of metrics and the evaluation of that data is an important function of Performance Engineering. Measuring time sensitive tasks before, during, and after training provides a simple measure of performance. Unfortunately, like almost any simple measures of performance, it isn't worth much out of context. Without an understanding of the context, it is impossible to attribute any change to training. One important reason why many training development people neglect this kind of in-depth analysis of results in context is that they know how to do what they are doing well and don't want to know if they will be required to do something different. Performance problems are almost never solved by training alone, just as they are almost never solved without training.

Very few managers would dispute the fact that training to a certification standard is the way to reduce the number of critical errors in complex processes. However, the moment to moment motivation of employees is beyond the scope of their certification training: Just because Mary can weld six unions an hour doesn't mean she will weld six unions properly when her child is ill or when she is mad at her supervisor.

The hallmark of the best training programs is that they provide a foundation for additional skills that were outside of the scope of the original training. All training ought to be designed to make the acquisition of more knowledge likely from the knowledge the student already has. There is a depressing amount of training with blinders on out in the real world - training developed so exclusively within its own defined competencies that it offers nothing to any companion curriculum. In measuring the effectiveness of training, it is important to measure these secondary impacts which lower the cost of subsequent training.

3.2 ROI by the Numbers

When looking at ROI and cost benefit analysis, it is important to remember that:

- Improving efficiency means achieving the same results with lower costs.
- Improving effectiveness means achieving better results with the same costs.
- It is possible to get better results with lower costs, and this is called improved productivity.

Scenario of a small training organization:

After discussing the training program with the managers of departments where the training occurs, although there are no directly observed statistics (or budget to collect them), everyone agrees that training program A has resulted in an increase in productivity of 22% compared to direct OJT by co-workers and supervisors. Whether the analysis is valid is not as important as the fact that it is defensible and mutually agreeable to the stakeholders. If the total output of the affected workforce nets the company \$1 million per annum, the training dividend is \$220,000. If the net cost of training is \$180,000 then the return on investment for training as a whole is 22.2%.

In this scenario, if the cost of an XML repository system is \$42,000, the return on investment for the initial year of operation will be 0%, provided that the same training gets produced as before. However, in subsequent years, it should be possible to produce more training and/or better training for the same regular investment. If the total output of this wider groups nets the company \$1.5 million per annum, the training dividend is \$330,000. You have increased net revenues by \$110,000 with an investment of \$42,000.

Scenario of a larger training organization:

According to Microsoft[1], if you have 100 employees using a system that cost you \$1,120,000 initially and requires a staff of 4 IT professionals to run it, the system must return an increase in productivity of 16% to break even the first year. Whether you spent this money on developing your own system or purchasing someone else's system, it is money well spent.

Collecting increased productivity from instructional designers depends on a number of factors:

- How much of the designer's time is spent actually developing content, as opposed to time spent developing curricula?
- How much of your content would actually get reused? In some organizations, the quantity might be almost nil and in others it could approach 50% or even more.
- How many repetitive training operations do your designers perform in order to get content to the students?
- Do your designers have the willingness and ability to change their methods?
- Does your organization possess the infrastructure and the commitment to design or customize, implement and use the content reuse system?

Assume that the instructional designers in your organization spend approximately 60% of their time accomplishing some aspect of content development. Further assume that of this time, roughly two-thirds would be generally unaffected by content reuse. If the content reuse system you implement results in that employee becoming 15% more productive in these tasks, then the net result for that employee is a 6% increase in productivity overall.

For every organization, there is some training that is mandatory, some that is essential, some that is preferred and some that is optional. Mandatory training represents those training hours that must be delivered to meet statutory or contractual obligations. Essential training provides to the employees the skills and knowledge they need to perform their jobs to a minimum standard. Optional training provides the employees with the skills and knowledge to excel. In today's tough economic conditions, many organizations have been trimming their training efforts to such an extent that they are beginning to see negative productivity results.

Faced with such realities, these organizations are faced with the challenge of providing a competitive, sustainable solution to obtaining quality training that facilitates excellence. An XML-based content-reuse system qualifies as an excellent example of such a solution.

3.3 Competing with Consultants

4 Conclusion



Even if you're on the right track, you'll get run over if you just sit there. – Will Rogers.

Implementing a robust content-reuse management system for the enterprise is not just an idea whose time has come, but an idea long overdue. Far too often, training departments must make hard decisions in hard times that end up being false economies. The underlying technology of XML has been proven in numerous settings over the past 20 years, first in government and then in the private sector. It is rapidly getting to be the case that if your enterprise does not implement such a system, you will compete at a considerable disadvantage in your marketplace.

In summary, the advantages conferred by the XML content repository and reuse system are:

- **Economy** - Provides the same economies of scale as automating any labor intensive, customized process.
- **Communication** - Provides a naturally unifying influence on the organizational communication; both internally and customer-facing.
- **Quality** - Provides the ability to attain higher quality levels and increased consistency across all training and similarly-sourced deliverables.
- **Productivity** - Provides additional capacity to produce mandatory training at a lower cost. Confers surplus capacity that can be used to develop more effective training that raises enterprise-wide productivity.

References

- [1] Microsoft. *Content Management Server, Return On Investment Calculator*. <http://www.microsoft.com/cmserver/evaluation/roicalculator.xls>, 2003.

Index

abstract, 1

case study, 7

cots, 5

custom xml solution, 4

home grown xml, 4

introduction, 2

roi, 8

unified content strategy, 3

xml off-the-shelf, 5