Reducing E-Learning Development Costs Using a Streamlined XML-based Approach

Bill Bandrowski, Emily Copp
Concurrent Technologies Corporation
Bremerton, Washington
band@ctc.com, castille@ctc.com

David Castillo
Buena Vista Learning LLC
Bainbridge Island, Washington
dcastillo@buenavistalearning.com

ABSTRACT

To maximize courseware development efficiencies, the project team integrated a suite of XML-ready commercial-off-the-shelf tools to streamline development resulting in a significant return on investment (ROI). This integrated toolset or "platform" supports a content development process that is nearly seamless from storyboard to final courseware. This proven approach can be replicated by other organizations to achieve a similar ROI.

Expressed in more technical terms, during run-time, a Flash-based e-learning engine combines XML-based content, "harvested" from MS Word storyboards, with a library of reusable interaction templates. Additionally, the e-learning engine provides support for configurable functionality such as navigation, assessments, and SCORM. Overarching development goals and methods associated with the platform include the following:

- Promote quality learning while maximizing instructional design investment through a well-designed, reusable interaction template library.
- Minimize manual manipulation of content and the opportunity for error focusing on single-sourcing.
- Maximize interaction template reuse by decoupling presentation elements from content.
- Facilitate XML editing with user-friendly forms.
- Maximize software programming investments by centralizing code and utilizing an object-oriented approach within an e-learning engine.
- Facilitate change management and reduce life-cycle maintenance costs by externalizing media assets in their native form.

Each development goal is supported with examples. Current platform components and workflow processes are described, and the significant ROI is summarized. The authors discuss lessons learned, alternative technologies, and feature enhancements —all potentially useful for the future. Concepts are illustrated by a federal effort to develop a Web-based training component for pharmacy technician training to be shared by the US Air Force, Army, Coast Guard, Navy, and Veterans Health Administration. The e-learning platform is being used for this large design and development effort (approximately 40 courses). Current data associated with this program point to an approximately 33% efficiency factor, as well as other areas of ROI.

ABOUT THE AUTHORS

Bill Bandrowski, M.Ed. is a Manager for Instructional and Performance Technology in the Bremerton, WA office of Concurrent Technologies Corporation, an independent, nonprofit, professional services company. He has over 25 years experience in the education and training field, with a focus on instructional technology development. He is the *CTC* Project Manager for the Pharmacy Technician WBT.

David Castillo is an author, solutions architect, and independent consultant who has worked with leading companies including Marriott International, Boeing, and USAA. With more than 20 years of experience designing and developing instructor-led and Web-based training, he served as the Concurrent Technologies Corporation's lead software architect for the Pharmacy Technician WBT.

Emily Copp is an instructional developer at Concurrent Technologies Corporation with over 10 years of experience in multimedia development, interaction design, instructional design, and usability testing. She contributed to the book Flash MX for Interactive Simulation, leading the writing on chapters dealing with device examples, case studies, and usability testing. She was instrumental in development of the Pharmacy Technician WBT.

Reducing E-Learning Development Costs Using a Streamlined XML-based Approach

Bill Bandrowski, Emily Copp
Concurrent Technologies Corporation
Bremerton, Washington
band@ctc.com, castille@ctc.com

David Castillo
Buena Vista Learning LLC
Bainbridge Island, Washington
dcastillo@buenavistalearning.com

INTRODUCTION

One of the most daunting challenges organizations face when producing custom e-learning content is to develop the capability to produce effective training in a streamlined and cost-effective manner. Typically, custom e-learning development, built on solid instructional design and current software development standards, is time-consuming and costly.

As a result, organizations sometimes default to less complex "slide show" courses or attempt to implement proprietary learning content development systems. What is needed is a development process that is nearly seamless from storyboard to final courseware allowing for development efficiencies that reduce production times from months to days, and reduce costs from being prohibitive to being reasonable. One solution to these challenges is based on XML implementation.

Purpose

There are many ways that available tools might be integrated to reach a solution similar to the e-learning "platform" (the platform) described in this paper. The purpose of describing this solution is to stimulate thought and open doors of possibility for program managers. If the decision is made to implement a similar platform, that decision should be guided by a clear understanding of not only the technology, but implementation issues associated with both components and process.

A second purpose is to provide insight for instructional designers and developers on effective repurposing of instructional design and associated software.

The paper presents six development goals and examples illustrating how the project addressed each of them. It then sketches the components and course development process supported by the platform and notes the return on investment (ROI). The paper concludes with a look at lessons learned, alternative solutions and feature enhancements, and some thoughts for program managers considering adopting the platform.

Background

This streamlined process was developed and implemented by Concurrent Technologies Corporation (*CTC*) in support of a collaborative e-learning effort involving the US Air Force, Army, Coast Guard, Navy, and Veterans Health Administration. The program is currently developing 40 courses (approximately 90 hours) of reusable pharmacy technician training for the five collaborating organizations. The Web-based training is one component of the various training approaches being employed by the five agencies for training pharmacy technicians.

The project has purposely "pushed the envelope" of instructional design, development processes, and methodologies in order to demonstrate best practices. Figure 1 shows a sample screen from developed courseware.

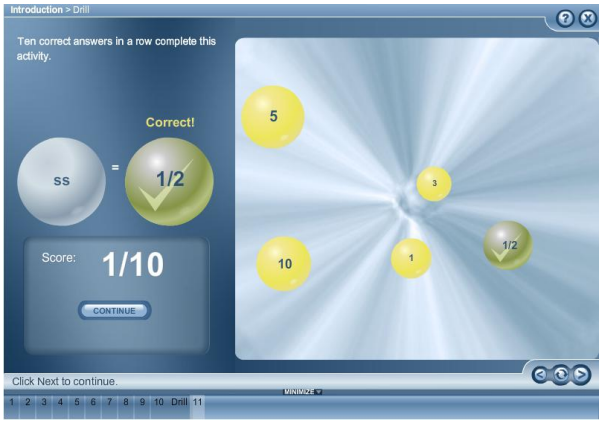

Figure 1: Sample Screen – Developed Courseware

Solutions presented in this paper build on earlier work related to 1) the challenges of separating content from presentation and, 2) the potential to use XML solutions

for streamlining development. Ideas were drawn from the work of researchers and developers including M. David Merrill (2007, 2008), J. Kaye and D. Castillo (2002), K. Mentor (2006).

The E-learning Platform

The project team integrated a suite of XML-ready commercial-off-the-shelf tools to streamline development. This has resulted in a significant ROI; current data point to an efficiency factor of approximately 33%, as well as other areas of ROI. The proven approach can be replicated by other organizations to achieve a similar ROI.

While some technical jargon is unavoidable in explaining the platform, the basic premise is this:

Keep all the major elements of courseware (content, instructional strategies, layout, graphics, interface, software functions, etc.) separate from each other until the very last moment of production, and avoid duplicate entry of any data. This makes it easier to update specific elements and reuse individual elements in various ways. It also results in a more efficient content development process over time.

A more detailed technical description of the platform (*E-Learning Platform Summary*) is available from the authors.

BUSINESS DRIVERS

Three critical business drivers were 1) high-quality training products generated by a cost-effective design and development process, 2) a design and development process, built upon the concept of an engine, that can be repurposed in similar projects, streamlining workflow without sacrificing the flexibility needed for innovation, and 3) a proof-of-concept that shows what might be done with instructional templates that capture the most effective instructional design strategies and reuse them in courses with comparable types of content.

These business drivers generated six development goals which then guided the allocation of resources on the project.

DEVELOPMENT GOALS

Each of the six development goals is illustrated with an example of how the project is working to realize it.

#1: Promote quality instructional design while maximizing investment.

For high-quality, effective e-learning, there needs to be a balance between 1) repeatedly engaging in original instructional design/development that results in duplication of effort, and 2) forcing content into pre-determined models of instructional strategy.

The goal of promoting quality instructional design while maximizing investment is supported by a library of reusable interaction templates. Each template evolves in the normal process of designing effective instruction for learners. However, all the elements except for the content are generalized, making it reusable with other content.

Taking advantage of these well thought-out instructional strategies did not limit the designers, since they were able to develop single-use interactions as well as introduce new templates that can be added to the template library.

Figure 2 displays a template from the library that can be used to present various information about parts of an object, a concept, or a process.

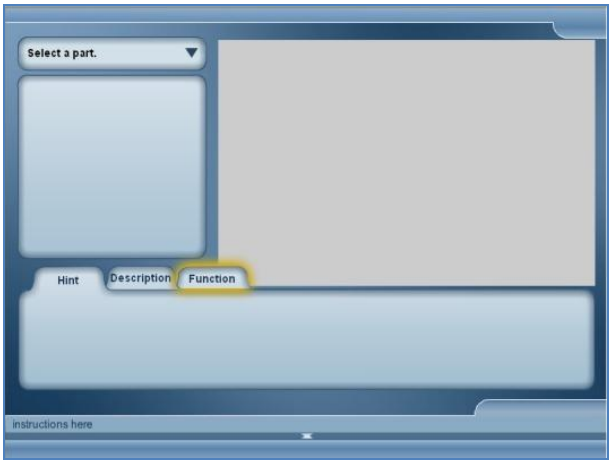

Figure 2: Interaction Template #1 – Parts of

Figure 3 displays a template from the library that can be used to present the steps of a process or procedure.

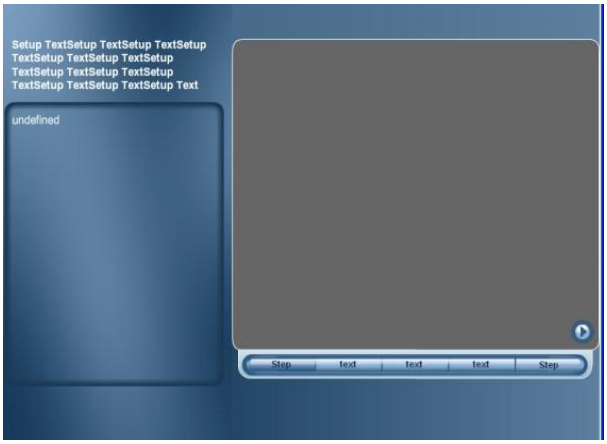

Figure 3: Interaction Template #2 – Process Steps

#2: Minimize manual manipulation of content and the opportunity for error.

One goal is to have a single design source for the development of course content, accessible for manipulation in the production cycle, and accessible for changes throughout its lifecycle. Ideally, changes made to design files will dynamically update finished courseware.

Working towards this goal, the engine is designed to combine individual elements as it generates the finished courseware. One of these elements is the storyboard, a critical design element, which differs from a traditional storyboard in that its content is exported to XML files that the engine can combine with an interaction template.

Storyboard Forms

Designers use storyboards to display raw content and communicate instructional strategies, directing how sections of content will be divided into screens, often called pages. The use of storyboards allows for review and approval of instructional strategies and final content choices without incurring the cost of developing the courseware.

The platform takes this concept of a storyboard and extends its usefulness by standardizing each instructional strategy as a unique "empty" storyboard form. Each storyboard form is organized to display a specific type of content, and is codified for reuse with similar types of content.

The storyboard forms, developed for reuse in the platform, allow all text content to be directly consumed by the e-learning engine, with no intermediary manual manipulation. Microsoft Word 2007 was used for the storyboard forms because of its XML-based foundation. Figure 4 shows a typical storyboard form prior to the addition of specific content.

Template: Flash Cards	File: flashCards	Page: 5
XML: flashCards	GUID: Click here to enter text.	Page Identifier:
Course: Click here to enter text.		Lesson: Click here to enter text.
Progress Check: Choose an item.		Remediation: Choose an item.
Topic: Click here to enter text.		Tab Topic: Click here to enter text.
Subtopic: Click here to enter text.		Tab Subtopic: Click here to enter text.
Display Mock-up (a representation of layout and image ideas)		
		
Graphics Description and File Name		
Activity Description		
Learner can toggle cards that have terms on one side and definitions on the other. Stack of cards to the left. When the learner drags the card over, they see the front. Learner must click to toggle the card before picking a new card or putting in the discard pile. Then learner can click and drag to the original pile or the discard pile or learner can select a new card and the middle card is automatically sent to the original pile. Learner has an option to shuffle the original pile at any time. "FLIP CARD" button below middle card.		
OST		
OST: Drag each item to its appropriate location.		
Drop Pile Instructions: Drag the top card to the center. Click Flip Card to view both sides. Drag the card to the discard bin when you have mastered the information.		
Instruction OST		
Pre-action: Drag each item.		
Post-action: Click Next to continue.		
Incorrect Feedback: Please complete this activity.		
Match Items		
Item	**Match**	
Item 1: Click here to enter text. Item 1 Image:	Match 1: Click here to enter text. Match 1 Image:	

Figure 4: "Empty" Storyboard Form

The fields that are grey-highlighted are content control fields. Some content (post-action instructions and incomplete feedback, for example) will be used every time this form is populated, and so, is already loaded with default data. Any field with the instruction "Click here to enter text" is a content control field that will "carry" instructional material unique to the page. The storyboard forms are organized in a library called QuickParts, a feature of Microsoft Word 2007.

#3: Maximize interaction template reuse.

Each interaction template is basically an instructional strategy. From an interface point of view, the template can be configured with any number of "skins," differing combinations of fonts, color schemes, interface elements, graphics, layouts etc. This is accomplished by separating content (XML) from presentation (Flash or HTML). From the content point of view, there may be more than one appropriate interaction for use with any particular piece of content. Separating content from presentation further encourages creative exploration, since an instructional designer has the

freedom to "audition" the content in multiple strategies before settling on the best one for the learners' needs

Another aspect of this reusability involves repurposing content for different audiences. For example, the content for an activity designed for a Flash drag-and-drop interaction could be automatically repurposed to work with simple HTML pages and radio buttons, making it readable by assistive technologies for vision-impaired learners.

A third type of repurposing, one that is rapidly gaining attention in the training world, is the ability to dynamically convert content for different delivery methods. For example, training designed and developed for standard laptop or workstation computers, can be repurposed for delivery via mobile devices. Doing so requires adjusting to the more compact layout and interface limitations of mobile devices, such as the inability to include rollover events. The separation of content from presentation elements facilitates this conversion to mobile devices.

Overall, the goal is to make use of opportunities that maximize interaction template reuse by employing the one-to-many concept.

#4: Facilitate User-Friendly XML editing.

XML, a language that allows the sharing of structured data, is a critical component of the platform. It allows content to be captured in a non-proprietary format that is readily consumable by emerging technologies.

Content XML: Page Information

Storyboard content is exported to content XML that can be read by the e-learning engine. Additional presentation attributes (final layout information, media file references, etc.), required to render the page are then added to the content XML. Chunking content at the page level facilitates content reuse since pages of content can easily be added or removed.

Sequence XML: Page Sequencing

Storyboards communicate the intended order of pages in a lesson or lessons in a course. The engine receives this information from an XML file—a sequence map. The sequence map communicates the correct ordering of pages to the engine and points to the associated interaction templates and content XML files.

Working directly in an XML document introduces the potential for error, since content is encoded with XML tags and must conform to strict rules. One development goal was to create less confusing, more user-friendly XML editing forms.

The platform uses Microsoft InfoPath to create both XML sequence maps and to add specific page attributes to the content XML, although other tools could be used such as browser-based forms. Figure 6 shows a portion of an InfoPath form specifying page attributes, or presentation information.

SETUP (optional)
The setup section is non-interactve text and images on the pages.

Text (optional)
Text called On Screen Text (OST) is content that gives direction to the user or is used on a non-interactive pages.

ID - Must be Unique	Src Text to be shown on screen	W	H	X	Y	Styles
ostSetupTxt	Drag each part to its appropriate location.	280	120	15	30	setupTextWhite

Images (optional)
Images in the setup section are used for background images and non-interactive images.

ID - Must be Unique	Src Image Name	Clip	X	Y
bgImage	0050_ddTextImageBg.png		310	0

Figure 5: Page-Level Attributes Detail

The engine makes use of this information as it combines the content XML from storyboards with interaction templates to generate the finished page.

The ability to edit XML without introducing errors saves program resources by putting those changes either 1) closer to the point of the change (the designer), or 2) as a task for the lowest-cost person capable of making the change (administrative staff).

#5: Maximize software programming investments

Considerable effort has been expended to design and develop the engine to be as flexible as possible in order to meet various e-learning design requirements and to maximize the investment in software programming representing a distinct savings over time.

For typical courseware development, the engine can be used in its present state without modification. Some of the functions currently supported by the engine are:

Navigation functions
- User interface components with support for new graphical schemes and layouts
- Multiple modes (presentation, remediation, and review)
- Progress bar
- Completion status
- Indication flags for progress checks
- Title and page topic support
- Sub-page navigation
- API support for alternative navigation schemes
- Lessons within a lesson (module) and SCORM 1.2 support

Assessment support
- Multiple interaction templates available
- Question weighting
- Scoring functions
- Question bank
- Question flagging
- Learning objectives and mapping
- Remediation support

Presentation and style support
- Fonts and size
- Bold and italics
- Bullets and numbered lists
- Color
- Special symbols
- MathML

Asset management
- XML import
- Graphic import
- Audio support with XML-based cue points
- Single-sourcing for content

SCORM support
- Communications with LMS
- SCORM 2004 objectives
- SCO sequencing support

Developer support
- Configurable modes for development, review, or production
- Dynamic versioning and titling
- Reviewer utility including review interface, database and reports
- Various development tools including debugging window, developer navigation, etc.

Modifications to the engine to support new features and functions would follow normal software development processes, including requirements definition, design, and prototyping prior to production implementation.

Each function is modularized as much as possible to facilitate this revision process. Any updates to the engine code should efficiently cascade through all training software facilitating quality assurance activities.

#6: Facilitate change management and reduce life-cycle maintenance costs.

Content revisions can be efficiently made throughout the lifecycle of the training by modifying easily accessible text and multimedia files.

Media Assets Repository

A critical component of the platform and training content is a media repository containing all the media assets associated with each lesson and course. These assets include the custom content, graphics, animations, and audio files incorporated into the interactions, as well as the reusable assets associated with each interaction template.

Developers create custom graphics and animations based on layout details established for each interaction template. Any single-use media assets are incorporated into folders designated for single-use interactions.

An additional asset repository holds all reusable media associated with all interaction templates used in the training.

COMPONENTS AND COURSE DEVELOPMENT PROCESS

The platform is a general framework that could be implemented using a variety of tools. The *CTC* solution consists of:

- SharePoint (permissions/version control, information policy management, document libraries and lists, and workflows)
- Subversion (software version/configuration controls)
- MS Word 2007 storyboard form library (Content controls, QuickParts)
- XML Schemas
- Visual Basic for Applications (VBA) scripts (exports content from storyboards)
- InfoPath forms for content XML
- InfoPath forms for sequence maps XML
- Media assets (graphics, videos, audio)
- Adobe Flash-based interaction template library
- Adobe Flash-based e-learning engine

When storyboards are finalized by instructional designers and are ready for development, content is exported from the storyboard into XML files. These XML files and other media assets are integrated by the engine at run-time, generating the courseware.

Process

1. The designer creates an instance of a storyboard document based on a Word template (.dot file) within the Subversion environment.
2. Within this storyboard document, the designer inserts a storyboard form from the QuickParts menu, one that is appropriate for the specific type of content being taught.
3. The designer populates the storyboard form content control fields with raw content.
4. If required, custom storyboards forms may be mocked up by the designer and inserted into the storyboard to communicate single-use interaction requirements or to be considered for template development.
5. The developer runs scripts to export the storyboard data and create content XML files based on schemas, allowing content to be joined to default presentation data in the XML files.
6. The developer uses InfoPath forms to add additional layout details and attributes to XML files such as graphic and audio file names.
7. The developer creates sequence map using InfoPath to define page sequence and provide page-specific information for each lesson and assessment, such as page number, topic and subtopic.
8. Assets such as graphics and audio files are created by the developer and stored within Subversion (controlled versioning environment). All assets including XML files are then added to asset repositories.
9. The following assets are integrated by the engine at run-time to render the final courseware:
    a. Lesson content XML files
    b. Assessment content XML files
    c. Page sequence maps – XML files
    d. Graphic and audio assets
    e. Interaction templates
    f. Single-use interactions

Figure 6 illustrates the internal process of the platform combining all development assets in the Run-time environment.

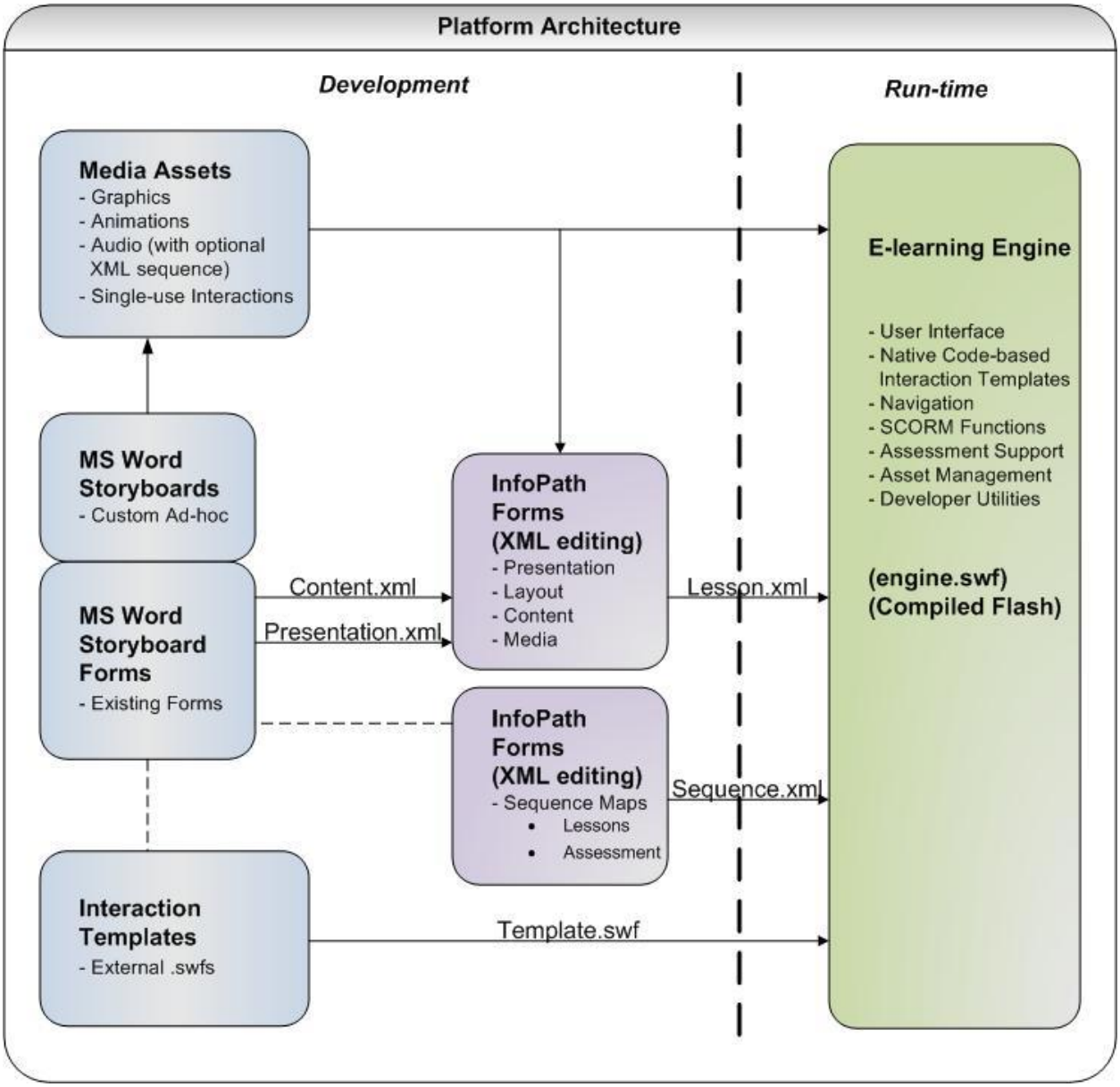

Figure 6: *CTC* E-learning Platform

RETURN ON INVESTMENT

Given the 30-month period of performance of the original project, detailed monthly data associated with development costs have been available and were analyzed. As expected, initial investments were significant. Also as was expected, the return on investment has been significant—costs reduced dramatically in later phases of the project. Because of the economies of scale (development investments spread over the 90 hours of content), final costs per hour have been reasonable and relatively low, and well worth the up-front project investment. But true ROI begins to be realized by repurposing the platform for the next phase of the project or completely new e-learning projects.

One new project that recently took advantage of the platform as the starting point is an example of this kind of potential ROI. Industry standard for the development of an hour of level 2-3 multimedia training ranges from 400-600 hours of effort. Using the platform and reusable interaction templates, the project designed and developed an hour of custom-built training in 300 hours of effort, a savings of 25-50%.

In addition to cost savings, recent projects that have used the platform have reduced development time from

a typical 6-month effort to a single month. Efficiencies are realized not only in decreased development time, but in organizing content, prototyping, storyboarding, graphics design, and quality assurance activities.

CONCLUSIONS AND NEXT STEPS

Lessons Learned

Three significant lessons learned were identified early in the project and prompted a re-thinking of the team's established approach. As the project evolved, mid-stream adjustments continued to hone the team's flexibility and creative use of the new tools and processes.

Just-in-Time Training
Anytime new tools or processes are introduced, there is a need to make sure all team members access and use them appropriately. The particular platform toolset that was used for this project did not arrive as an integrated, intuitive application out of a box. In fact, decisions about what to use and how best to use it were in flux for some time. This is by no means unusual in the world of software design, since technology changes at a dizzying pace and new functionality drives innovation and vice versa.

The need to plan for just-in-time training will continue to be critical in relation to the platform, since new technologies may provide better solutions for the processes the platform currently supports, and any new tools will impact the nature and workflow of the processes. A team learning curve that is longer and potentially steeper than for traditional projects should be expected.

Integrated Team
Traditional models of training design and development often emphasize the importance of compartmentalizing the designer and developer roles, assuming that developers take the designer's work and render it as indicated in the storyboard. Developers have been stereotyped as reclusive, focusing on interactions with fellow-programmers. Designers have been seen as demanding and inclined to assume that anything designed can and should be developed.

While most teams are probably more cooperative than these stereotypes would predict, there is nevertheless an important lesson learned: the platform requires more close cooperation and team work from *all* parties than a more traditional approach would require. Discussions about potential new interaction templates must begin the minute the types of content are known since storyboards are dependent upon many decisions being reached about the interaction template that will flow from it.

In order to communicate, not only must everyone be trained on the new tools, they must speak at least some of each other's language and understand the implications of their demands on workflow, reviews, and quality assurance activities.

Creative Tasking
As the team became familiar with each new tool in the toolset, it became clear that some of the tasks required a lower level of expertise to accomplish than would have been expected using a traditional approach. For example, designers were able to make text changes directly in the XML files, eliminating the need for assigning them to a developer and tracking the change request.

In addition, creating the sequence maps no longer required a person trained in programming; with the InfoPath form serving as a user-friendly interface, administrative staff were able to input the data, check for accuracy, and correct for error if need be, all without "touching" the actual XML.

Whenever possible, the toolset was implemented to realize this kind of time and cost savings.

Alternative Solutions and Feature Enhancements

The particular toolset that *CTC* uses for the platform is neither proprietary nor unique. The speed at which technology changes suggests that other emerging tools should be considered. They may prove even more efficient.

Below are a few alternatives and possible areas of improvement associated with the platform:

- Though InfoPath-based forms provide a user-friendly interface, other applications could support forms-based input while maintaining the integrity of XML schemas. One example would be form services from SharePoint, allowing editing of XML with a Web browser.
- Support for additional accessibility requirements per the Section 508 amendment to the Rehabilitation Act of 1973
- Metadata support
- XML schemas enhanced to adhere to standards such as S100D, QUIZML, Math-ML, XHTML, and XAML.

- Support for efficient simulation and level 4 interaction development

Adoption Considerations and Recommendations

Managers who are considering adopting the platform need to carefully weigh benefits against risks, and investments against returns.

Significant software engineering is associated with the following components: instructional interaction design, XML schemas and forms support, storyboard forms creation, and e-learning engine development. In addition, process improvements require major support for analysis, standardization, and implementation in order to realize envisioned efficiencies.

While there are clear benefits associated with platform implementation, risks are inherent with any software development activity and this one is no exception. Potential impacts on scope, schedule, and budget need to be considered.

What is required is a clear understanding of the organization's vision, both short- and long-term, for e-learning. This might include the volume of courseware to be developed, the criticality of production times and rates, and the technical expertise of the organization's design and development teams.

Adoption of the platform is recommended for those organizations that require:

- Relatively high volumes of courseware development, and
- Quick production turnaround.

The investment in a design and development team with the caliber of expertise required to work in this environment must also be considered, whether in-house or contracted. If in alignment with the organization's e-learning vision, the technology innovation such a team can deliver has the potential to translate into valuable business solutions.

ACKNOWLEDGEMENTS

Many dedicated professionals from Concurrent Technologies Corporation (*CTC*) as well as the five federal agencies have contributed to this ground-breaking collaborative Web-based training development project. Personnel from the US Air Force, Army, Coast Guard, Navy, and Department of Veterans Affairs, working with *CTC*, used the platform to produce engaging, user-friendly courseware that will play an important role in training pharmacy technicians within their organizations.

REFERENCES

Concurrent Technologies Corporation. (2009). *CTC e-learning platform summary*. Bremerton, WA: Author.

Kapp, K.M. (2003). *How long does it take? Estimation methods for developing e-Learning*. Retrieved spring 2009 from the ASTD Web site: http://www.astd.org/LC/2003/0703_kapp.htm

Kaye, J. & Castillo, D. (2002). *Flash MX for interactive simulation*. Clifton Park, NY: Delmar Thomson Learning.

Mentor, K. (2006). *Director and SCORM 1.3 SCORM SCO presentation engine (S2PE).* Retrieved fall 2007 from the Adobe Web site: http://www.adobe.com/devnet/director/articles/director_scorm.html

Merrill, M. D. (2007). A task-centered instructional strategy**.** *Florida State University, Brigham Young University-Hawaii, and Utah State University Journal of Research on Technology in Education*, 40(1).

Merrill, M.D. (2008). *Learners in a changing landscape: Reflections from a dialogue on new roles and expectations*. Visser, J. & Visser-Valfrey, M. (Eds.). New York: Springer Science + Business Media B.V.