

ko+kim

CONSULTING

# XML and Automated Publishing

By

Ben Ko, *Principal*

April 10, 2007

# What does automated publishing mean?

- shared standards
- clearly-defined workflows
- right tools for the right job
- software development
- process change
- survival

# Business drivers behind automated publishing

- Reduce time to market
- Release multiple products simultaneously
- Enable creation of new products that are too expensive to produce by conventional means
- Pressure to lower existing production costs
- Increased demand for customizing content

# Why XML is important to automated publishing

- Designed to be easily transformed into both print and electronic products
- Application neutral-format. Doesn't care what your layout engine is.
- Enables customization and versioning of content
- Improves ability to manage, search, and re-use content

# Case Study: custom publishing

- Enable customer to select content via web browser and create custom book rendered to HTML and PDF in real time

# Solutions

- HTML development decisions were relatively straightforward: XML to HTML via XSLT
- XML to print PDF decisions, both technology and workflow, required much more thought

# Key Requirements for Rendering Solution

- Supports XML in and out
- High quality rendering of both simple and complex layouts typical of college textbooks
- Leverage existing expertise in current interactive workflow in new solution
- Leverage new solution back into existing interactive workflow to make existing interactive workflow more automated

# Rendering Solutions considered

- XSL-FO (RenderX)
- 3B2
- XPP
- InDesign+Typefi
- QuarkXPress Server+Autopage

# XSL-FO

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

- RenderX's XEP is an XSL-FO processor engine takes input in XML, applies an XSL transformation to build XSL Formatting Objects representation, and then formats the Formatting Objects into PDF or PostScript.

# XSL-FO example code

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

# XSL-FO: Pros

- Growing market support
- Supports XML as input and is XML-based
- Affordable
- Both free and commercial FO-engine implementations available
- Community support

# XSL-FO: Cons

- Not yet capable of supporting the more complex layouts required by college textbooks
- Results cannot be leveraged back into workflows where designers and paggers need to manually modify the result
- Designer and formatter/programmer are working in different applications

# 3B2

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

XyEnterprise's XPP and PTC's 3B2 (now Advanced Print Publisher) are competing automated publishing engines that handle complex layout requirements for printed publications such as STM (Scientific/Technical/Medical) journals, legal documents, technical documentation, catalogs and directories.

# 3B2 example code

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

# 3B2: Pro

- Supports XML in and out
- Supports industry-standards, e.g., CALS, MathML
- XSL-FO support
- Supports XPath and Perl
- Solutions have been on market for ~20 years
- Supports batch pagination and some level of interactive pagination
- Incredibly programmable
- Fast batch performance

# 3B2: Cons

- Difficult to use directly by designers, introducing split into workflow between design program and composition program
- Requires significant engineering skills to leverage — engineering skills that are largely absent in case study vendor pool, thus requiring significant training (and hiring) at vendor pool or complete migration to different vendor pool
- Acquisition cost

# InDesign CS2 + Typefi

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

TPS consists of a workflow component, content management component, and automated page layout component. The page layout system drives InDesign to layout pages "with no scripting requirements at all" (Typefi).

# InDesign CS2+Typefi: Pros

- Supports XML in and out
- Supports DTD validation
- Supports XML to style mapping
- Used by some book publishers to automate page layout
- Page layout application can use the design template directly
- "Scriptless solution"

# InDesign CS2+Typefi: Cons

- Sophistication of page layout algorithms and support for complex layouts not at level of competing solutions
- Required customer to map their XML to Typefi's "Content XML" vs. natively supporting customer's XML
- Adobe had not yet released InDesign Server so only InDesign desktop solution was available

# QuarkXPress Server + Autopage

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

Autopage is an automated publishing solution for long documents in QuarkXPress. It automatically pages trade books, textbooks, directories and yellow pages including placement of art (ads), inline art, margin elements and footnotes while balancing pages with varying layouts.

# Autopage Code Sample

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

# QuarkXPress Server + Autopage: Pros

- Supports XML in and out
- Supports industry-standard CALS markup
- Supports XPath and Perl
- Autopage solution on the market for ~20 years
- Autopage solution produces layouts with fewer page faults than other pagination systems due to unique support for backtracking during pagination

# QuarkXPress Server + Autopage: Cons

- Supports complex math rendering but does not natively support MathML
- During interactive composition, supports checking for well-formedness but not validity
- After initial pages are composed by mapping XML to layout objects, further layout changes are driven by layout object rules only, not by XML (XML can be modified and re-run).

# Rendering Solution Selected

- QuarkXPress Server + Autopage

# Reasons

- Superior pagination quality
- Many composition vendors have experience with it.
- Autopage requires skilled composition experts but not software engineers to implement
- Leverages the QuarkXPress environment so no change required from design file to paged file
- Can be leveraged in interactive (QuarkXPress) and automated (QuarkXPress Server) environments

# Reasons

- Although it was behind some of the other applications in certain areas, e.g., MathML support and support for nested tables, these were not deemed critical for release and could be addressed with enhancements at a future date

# Customer Quotes

- "KyTek has 20 years of experience developing sophisticated page layout algorithms that is difficult for anyone else to match." from 3B2 integrator
- Keith Erf (creator of Autopage) "is one of the best programmers in the country." from Autopage's first customer
- "Autopage delivers more finished pages than the well-known proprietary batch pagination system we were using." from current Autopage customer

# Why QuarkXPress Server, not just QuarkXPress

- Automation advantage: Server is driven by HTTP protocol instead of interactive desktop GUI
- Performance advantage: Removal of overhead of GUI and screen updates improves performance
- Performance advantage: Able to divide jobs across multiple processors
- Reliability: 24/7 uptime
- Flexibility: Acts as backbone for many programmatic solutions, not just as designer's interactive desktop tool

# QuarkXPress Server + Autopage screenshots

# Paginate Dialog

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

# Pagination Parameters

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

# Pagination Parameters

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

# Pagination Parameters

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

QuickTime  
TIFF (Uncompressed  
are needed to see

# QuarkXPress Server Status Monitor

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

# Example HTTP request to QuarkXPress Server

- <http://localhost:8080/autopage/pccRH042005?xmlFile=&library=XServe:QuarkDDS:Documents:CustomLibrary&autopageRules=G5-2500:april5demo:param041705.jd&qxpOut=G5-2000:QXP:.qxp&jobFolder=SHAREWITHMAC:testPCCindex:&jobID=3942-2000&chapterID=&firstChapter=0&firstPage=01&numberStyle=1>

Automation is not just about  
the rendering engine

# XML is the cheapest technology and also the most expensive

- It's one thing to write an XML specification. It's another to create a workflow whose schedule and budget can support that specification
- Valid XML is not enough
- An XML workflow usually implies process change, so you need management support and budget support
- XML is ideally deployed at the front end, not the backend.

- XML is an enabling technology that can transform publishing processes but not if you use the XML to re-create the status quo

# Closing Comments

- XML has re-invigorated what publishing companies want to and can do with their content
- Not all companies are ready for the process change
- Due to economic conditions, many publishing service vendors today do not have the technical expertise or high margin revenues to be innovators to help the publishers establish best practices. Publishers need to take the lead and make their own investments

# Contacts

- XML workflow consulting, project management, training, and publishing software development
  - [bko@koandkim.com](mailto:bko@koandkim.com) ([www.koandkim.com](http://www.koandkim.com))
- QuarkXPress + Autopage automated publishing solutions
  - [sales@kytek.com](mailto:sales@kytek.com) ([www.kytek.com](http://www.kytek.com))
- QuarkXPress Server
  - [dliles@quark.com](mailto:dliles@quark.com)  
<http://www.quark.com/products/server7/partner.html>