

S.S.P. Tech Blitz

Thursday Nov. 18, 2004

1:00 pm to 2:00 pm: XML and the Production Process

“Journal Publishing in XML”

Peter Butterfield
Partner, PubLink Inc.

Journal publishing in XML

Driving the Composition process with XML.

- Is it always the answer?
- Is it ever the answer?
- Well, when is it the answer?

Scope of this presentation: Characteristics & benefits for the decision-maker, rather than technical details.

Peter Butterfield Contacts:

peter@pub-link.com

www.pub-link.com

Voice: +41.21.807.0617 (Switzerland)

Fax: +41.21.807.0618

Outline of Presentation

1. The journal publishing process, and where XML fits into it.
2. How XML adds structure to a doc. How it can drive effective processing.
3. Differences between SGML and XML for publishing.
4. Benefits of XML for print: The big picture.
5. Should you use XML? Pros/cons for your workflow.
6. XML tools: Categories, what's available in each, commercial and open-source.

1. XML in the journal process

- ? Document flow in publishing a journal issue
- ? Where XML fits in the flow

2. The XML document drives processing

- ? What XML is.
- ? How XML adds structure to a document.
- ? How the XML document can drive effective processing better, faster and cheaper.
- ? But not all XML is equal!

2. XML is: A method of structured mark-up

- ? XML is an encoding method for describing a document in terms of its *structure*.
- ? *It's machine-assisted mark-up.*
- ? *An XML document is in plain ascii: Text, mark-up and all. Any person or program can read it.*
- ? *Other mark-up or tagging methods describe in terms of appearance or presentation.*
- ? *How structured mark-up drives all subsequent uses of the document.*

2. XML works by: Tags and entities

- ? Every change point in a doc is tagged.
- ? Tags in plain ascii, within “<” and “>”.
- ? Tag is a name describing *type of content* that follows, not its appearance.
- ? The content piece has a closing tag that matches the opener. So, every piece in doc is discreet and operable.
- ? Tags may have *attributes following the name, allowing unlimited info about the content piece.*

2. XML works by: Tags and entities (2)

- ? Tag tree: Doc starts with one top-level or root tag, all others are branches.
- ? Tags recur in different tree branches, or contexts. In same way, whole branches can recur in different contexts (nesting).
- ? Tree of tags forms the document structure. So, the structure you see in a well-formatted, printed article is mirrored exactly in the tag tree.

2. XML works by: Tags and entities (3)

- ? Each content piece, positively and uniquely identified by tag-name and tree position, can be processed in any way.
- ? Content includes non-printing pieces: Notes, metadata, optional inserts.
- ? Non-standard characters are in plain text too, as *character entities* or *numeric character references* (typically Unicode).

2. XML works by: An external model (maybe)

- ? “Well-formed” XML has no external model; a tree of tags, and entities, as just described. Works great.
- ? A DTD is an external model, or rigorous description of all tags, entities and contexts. Each doc is validated against it.
- ? Schema: Similar to DTD in idea, different in construction. More widely used in web apps than print publishing.

2. XML drives: Unlimited applications

- ? Multitude of programs to transform XML doc into a useful end result.
- ? A set of protocols, from same org, to assist: XSL, XSLT, XSL-FO, Xquery.
- ? A wealth of tools, commercial and open-source, available (last part of presentation).

2. XML drives: Unlimited applications (2)

- ? Page formatters: Use XSLT to produce XSL-FO and pages.
- ? Desktop publishing packages all take in XML.
- ? Commercial composition systems *should all do XML: Import, or apply stylesheet of composition against it.*
- ? *Web publishing: XSLT to XHTML, to another XML or to HTML.*
- ? *Content management: XSLT or Xquery to extract pieces, as from a database.*

2. XML variants: Endless

- ? It's a protocol, not a fixed language. The first XML you see is probably someone else's.
- ? Existing DTDs: Only some are designed for scholarly journal articles.
- ? XSL-FO is an XML, but bad for general use.
- ? WordML is built to describe MS Word docs.
- ? Homegrown DTD or tag set: May be right answer, but take time to get it right.

2. XML variants: Endless (2)

What you want in an XML tag set or DTD:

- ? Sufficient complexity to handle all variants of content in your journal family.
- ? A minimum of useless tags.
- ? Structure of your journal articles should be mirrored clearly in structure of the tag tree.

3. What about SGML instead?

- ? For publishing, XML & SGML are pretty equivalent.
- ? SGML, the original, developed with full generality for any database application, of which printing was one.
- ? XML is a form of SGML with restrictions that focus it on rich document structure to be used on the web (initially). In XML, the optional features are at a minimum.

3. XML advantages over SGML

These may or may not apply to you:

- ? Don't need a DTD, can use more flexible well-formed XML. But validated data may be needed for parallel use of your data.
- ? Your composition engine may not be able to validate against a DTD.
- ? XML data more easily re-purposed to other non-print uses using XML tools.

4. Benefits of XML for print: The big picture

The XML method is:

- ? Well-tested
- ? Stable
- ? Venerable: 1.0 used for publishing since at least 1998. Predecessor SGML long before.
- ? Rich in the proven DTDs available. Most DTDs are free to use, and eminently adaptable to any scheme. Use them; they embody lots of development by lots of smart people.

4. Benefits of XML for print: The big picture (2)

(continued) The XML method is:

- ? Processable by lots of tools, for lots of uses.
- ? Information-rich: Using tags & attributes, all info about the document may be stored within it. Examples:
 - Info about a content piece, incl. comments
 - Metadata to guide workflow
 - Processing history
 - Field info for database storage/retrieval
 - Processing instructions

4. Benefits of XML for print: The big picture (3)

(continued) The XML method is:

- ? Great at standardizing pi characters of all kinds, with character entities.
- ? The preferred vehicle for documents with Unicode.
- ? An answer to re-keying, and the errors introduced there.

5. Should you use XML?

Your workflow may work well now. Will adoption of XML into the flow let you work better, faster or cheaper? Weigh these factors:

- ? Will it increase your efficiency or productivity?
- ? Do you want other uses of the data beyond paper publishing?
- ? Do you *need any*? For instance, if another org asks for your data, they may require XML.
- ? Will you keep your data for future uses? Be aware that the standard of XML may be expected in future.

5. Should you use XML? (2)

Calculating a change-over. In setting up an XML workflow, the main work is up-front, in building and perfecting:

- ? *The DTD or tag set. Even if you acquire it, there's a learning curve.*
- ? *The WP – to – XML translation application. This is always needed, unless authors are in-house using your XML authoring tool.*
- ? *The XML – to – Typeset-coding stylesheet.*
- ? *Any other XML – to – web or similar applications.*

Once that's done, you're off and running! Production through to pages can be more hands-off, even totally with a capable composition engine.

5. Should you use XML? (3)

Once that's done, you're off and running! Production through to pages can be more hands-off, even totally with a capable composition engine.

- ? Corrections and re-processing should be minimized, due to reliable document structure and bulletproof stylesheet. Page it right the first time.
- ? Corrections should be limited to: Improve line breaks, page breaks, figure/table placements.
- ? Future projects of similar style? Minimal adjustments to translation tables and stylesheet.

Conclusion: XML a seriously good choice for ongoing jobs whose design won't change very much.

6. XML tools

- ? Authoring tools: OpenOffice.org, Office2003
- ? Editing tools: Arbortext Epic, ...
- ? Word-to-XML conversion tools: ExTyles, Exegenix, Apex, others. MS Office 2003.
- ? Inter-XML transformation tools using XSLT: Between DTDs, to schemas, to XHTML, etc

6. XML tools (2)

- ? Page builders based on XSL-FO: Antenna House,
- ? Page builders, desktop stylesheet-based: InDesign, Quark, Frame, ...
- ? Page builders, commercial stylesheet-based: Miles, PubLink, 3B2, XyVision, Frame, ...