Good morning. My name is Ken Brooks. I'm President of Publishing Dimensions - formerly VP of Digital Content at Barnes & Noble, Inc. Publishing Dimensions provides innovative solutions to assist publishers in generating revenue through evolving digital content and rights channels. This usually involves conversion of pBooks, PDFs or Quark files into eBooks and setting them up at Peanut Press or LSI, but recently it's more and more involved with XML: using OEB, ONIX, XrML and other forms of XML in production and distribution.

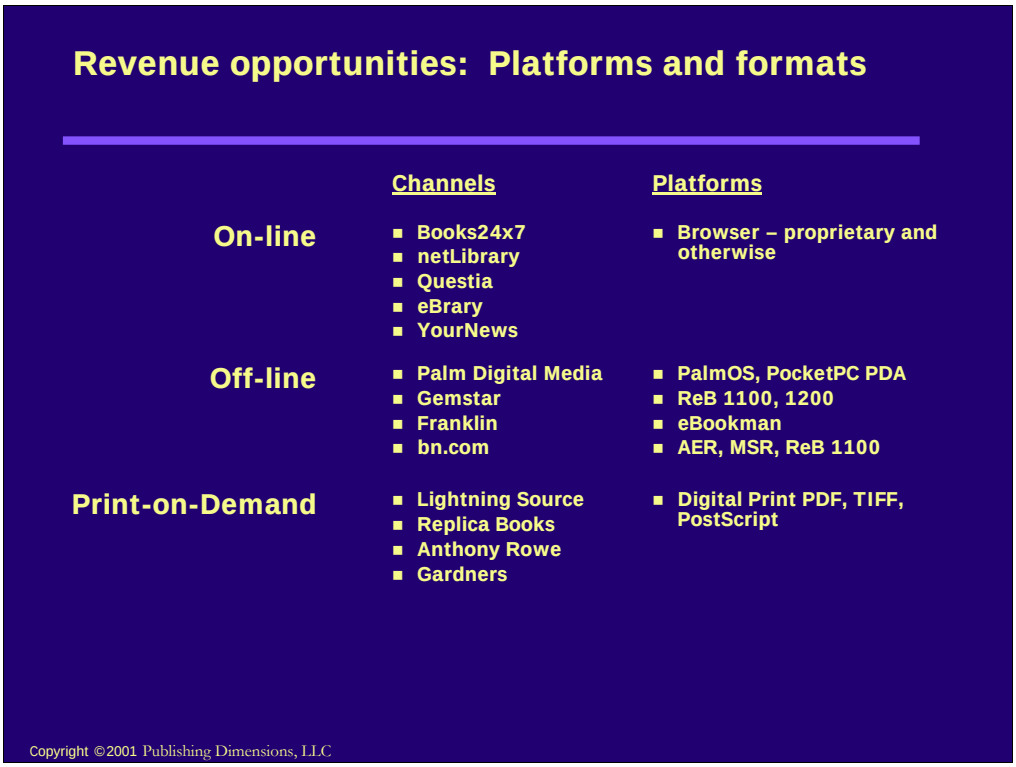
Subject: So that's what I'm going to address today: XML as it is in use in publishing, particularly book publishing, today.

Importance: I think you'll see that XML offers a pretty significant opportunity for revenue enhancement and cost savings in most aspects of your business.

Preview:

1. I'm first going to discuss some trends I'm seeing in publishing that are leading toward adoption of XML
2. I'll then talk about its use in content and metadata and wrap up with a discussion of areas to watch
3. Katriel Reichman from LiveLinx will then go into live examples of how XML actually works in these applications

Transition: There are a number of trends in publishing that are driving adoption of XML – overall publishing is getting more difficult.



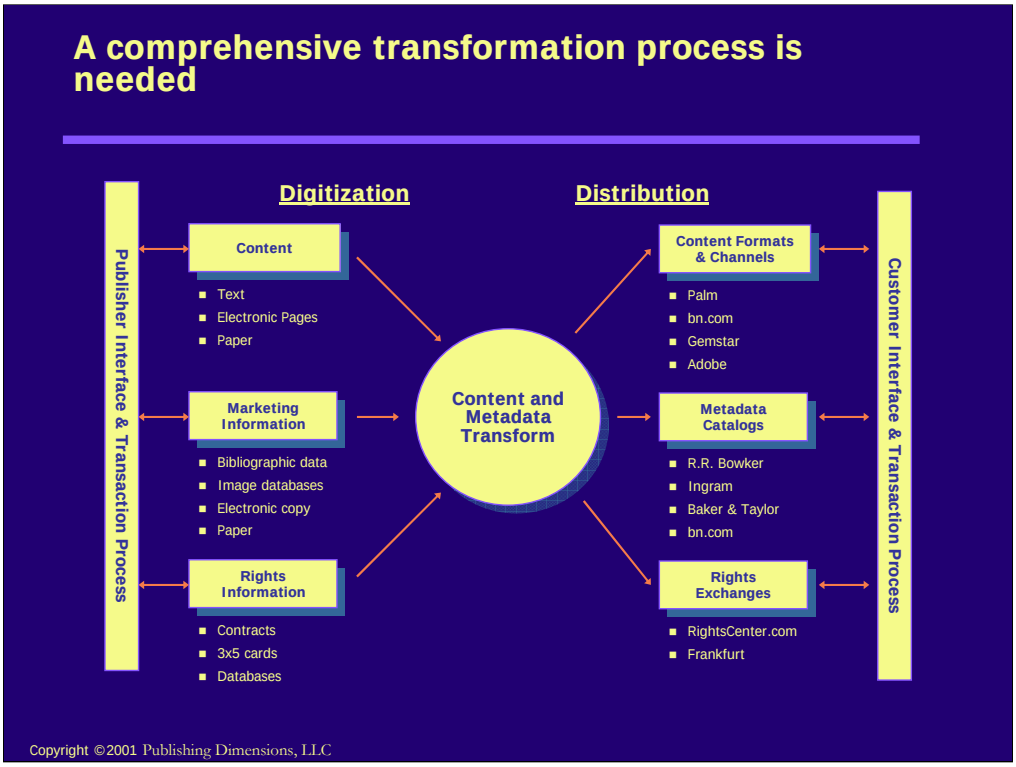
Reading platforms consist of combinations of hardware, operating system and reader applications. An example of a platform would be the a Windows PC running Acrobat, Glassbook or the Microsoft reader (that's 3 separate platforms) or a PocketPC PDA running either the Microsoft Reader or the Peanut Press Reader. A final example before we move on would be the Gemstar platform consisting of device and reading application or Franklin eBookman also running its own application.

Transition: Naturally with the constant changing of the channels and platforms comes a great deal of format instability

Format churn is endemic

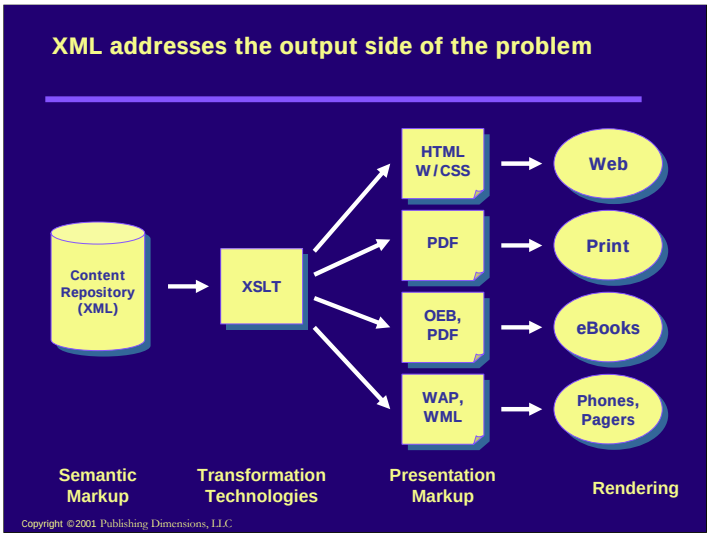
Format Family	*Variant*
■ XML-based	■ DocBook, DocBook Lite ■ TEI, TEI Lite ■ OEB, RHI Public Implementation ■ OEB, netLibrary spec ■ Gemstar (RB, SB, ReB 1100, 1200) ■ MSR (.LIT) ■ Palm Digital Media (PML) ■ MobiPocket
■ PDF	■ AER optimized ■ Standard PDF ■ POD PDF

Transition: This leads to a necessity of re-using / reformatting content a lot



Text	Straight text	ASCII
	Structured	HTML, XML, OEB
	Styled (formatted)	RTF, Word, WordPerfect, typesetting (3B2, etc.)
	Structured and styled	XML w/CSS, HTML w/CSS, SGML
Elect. Pages	Application files	Quark, InDesign, PageMaker, Frame, Ventura, LaTex
	PDF	
Fixed Pages	Hardcopy	Books,
	Scan	TIFF, PDF-Image
	Other	
	Microfilm, printing film, etc.	

Transition: XML is really the key to get there



I'm going to leave the conversion ***to*** XML (digitization) aside for the moment. That's something that Publishing Dimensions and others handle every day

Rather, let's focus on what happens once the content and metadata is in some sort of XML. The fact is that once you have your content and metadata expressed in a rich markup, it's relatively easy to move it into other formats – including other dialects of XML

Introduce XSL

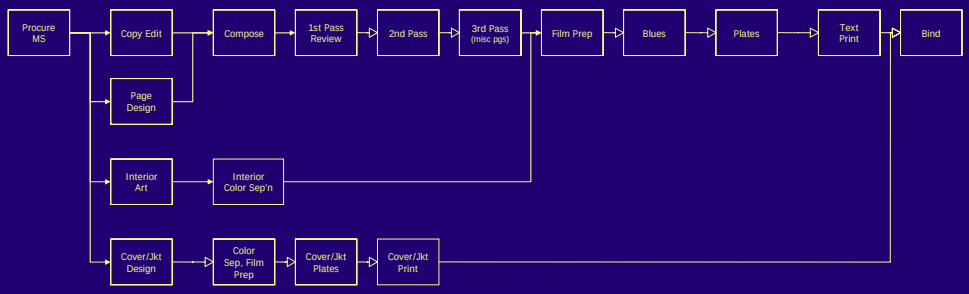
Tools are more prevalent, some being not terribly expensive and quite useful – primarily I'm talking about XMLSpy which is available under $200 per copy

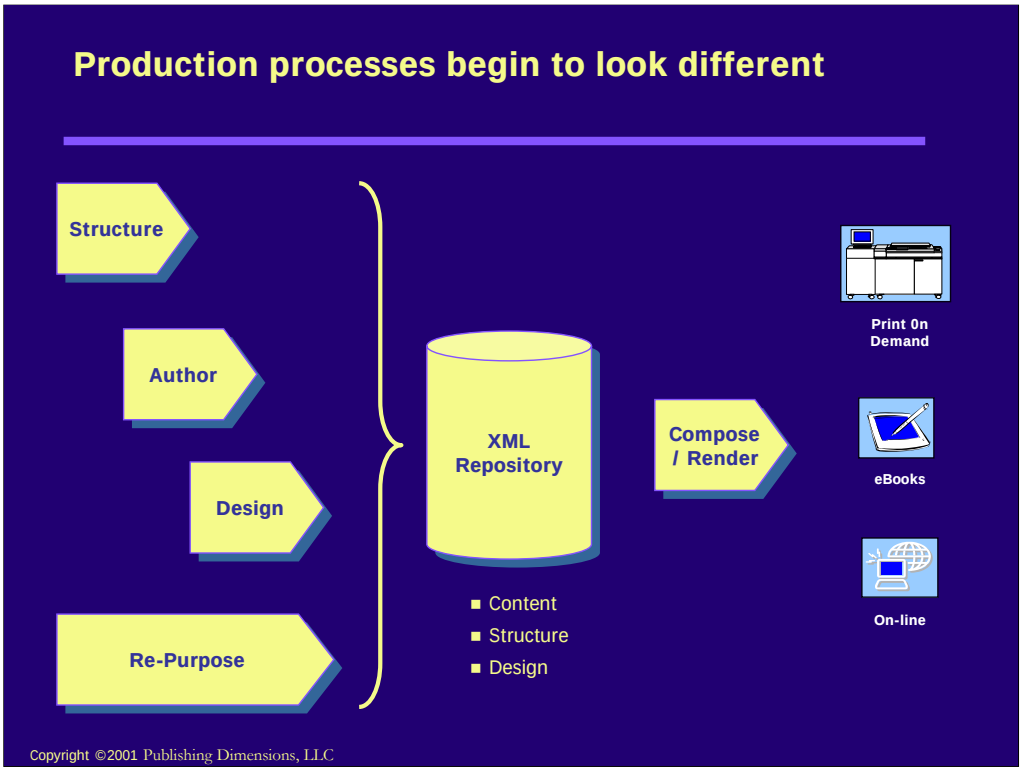
Many publishers are beginning to adopt XML for a variety of applications – both archival and presentation. Higher ed and STM publishers have used XML (or SGML) for a long time to archive their materials. Trade publishers are really just now getting into it with OEB.

Finally, nothing proves the point better than the number of applications that are out there in actual use. XML is becoming a serious transaction management standard to rival EDI – witness Microsoft BizTalk. OEB and other XML DTDs, along with XSL, are being used at many publishers to generate multiple title formats, and a newer standard, ONIX, is being used to govern the exchange of title metadata between publishers and their retailers.

Transition: So now we've seen that XML offers a solution, let's get into some more detail on XML as it is used in content markup. The first thing to realize is that books are really 3 separate things rolled together.

A generic print book production process...





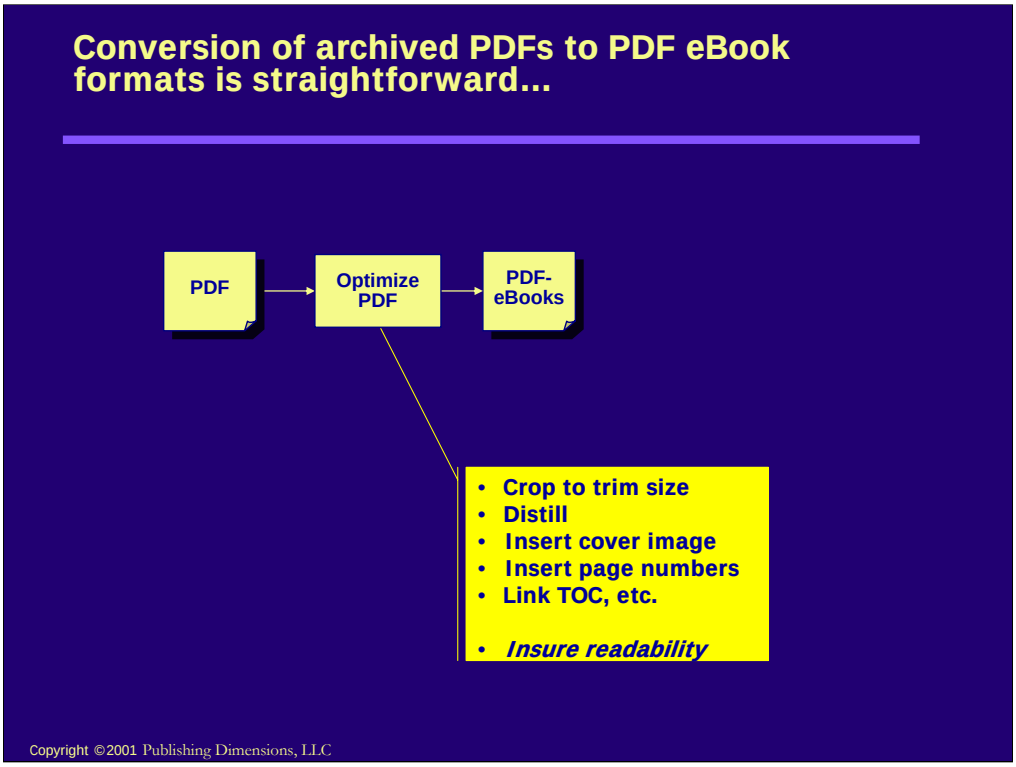
The key is to realize that you have three separate elements to contend with: content, structure and design. Each of these have processes associated with them. These are also offset in time from each other. Structuring usually starts first as the author and editor work out the outline. Then comes content development and finally design.

Re-purposing cuts across each component and involves extraction of each of the three elements. This is the role of the many conversion vendors out there, including Publishing Dimensions, that specialize in such processes.

These development processes flow into three "virtual" databases – probably contained within the same document management system. To get final output the content then comes together with format-specific designs, usually contained in XSL and CSS stylesheets, in a composition/rendering process.

At a high level these feed the three major branches of ePublishing: print-on-demand, eBooks, and on-line access. There's clearly much more to this than this high-level diagram, but it's a good way to think of the overall process.

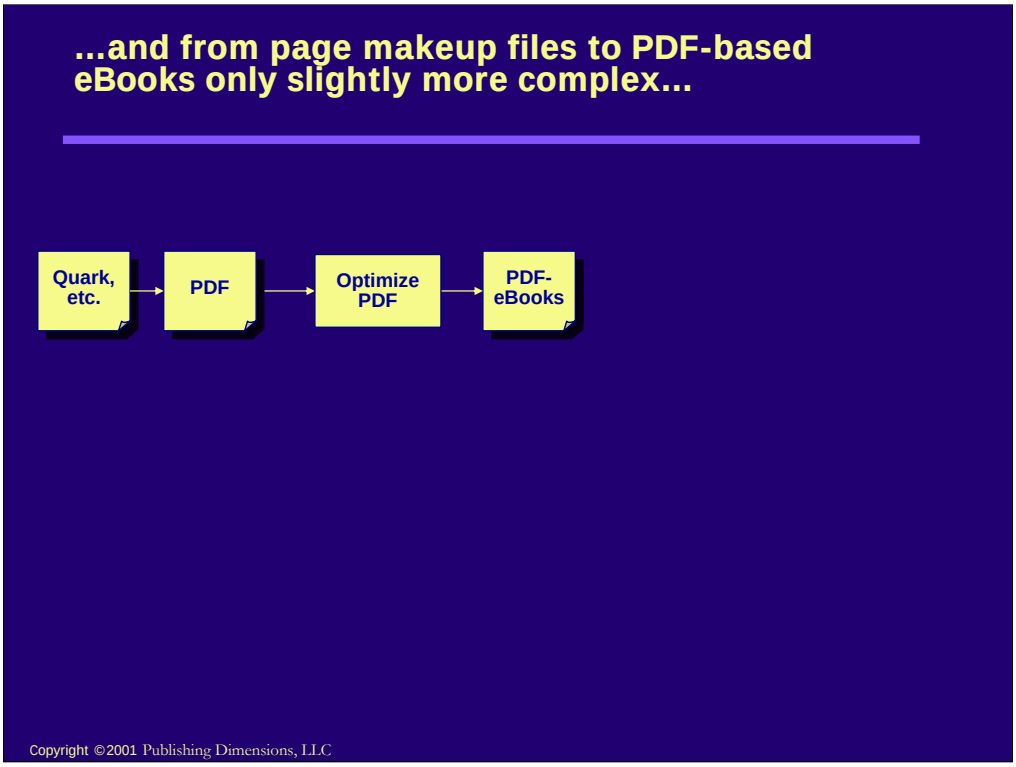
Transition: Let's now move into metadata. The key pieces are marketing metadata to populate industry catalogs and rights metadata to inform both rights sales and DRM solutions.



I'm illustrating this process with this very straightforward, indeed boring, flow chart. You start with an archived, high-resolution PDF generally with crop marks and possibly split into chapters or forms. You put them all together with Acrobat or one of its plug-ins, then optimize the PDF by cropping to final trim, distilling using the eBook settings suggested by Adobe, setting page numbers and linking the Table of Contents. All of this is done in Acrobat. You can also generate bookmarks, thumbnails and the like within Acrobat, as well. You then attach a cover image to the title – created very easily in Photoshop – and you're done!

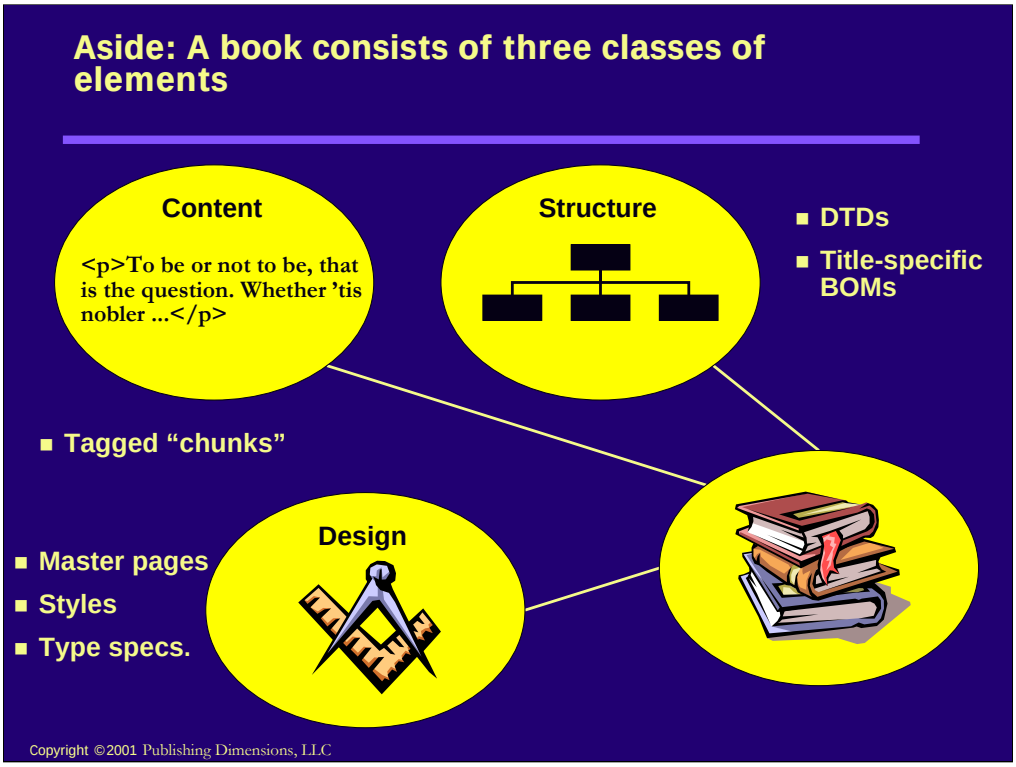
As reflowable PDF-based eBooks become popular, I imagine that we'll probably add some more tagging in here, but the accessibility tools offered through Acrobat 5.0 look likely to provide some help there.

Transition: If you're starting with files in InDesign, PageMaker or Quark the process is only slightly more complex…



It requires only that you "print" the PDFs first – something that's becoming quite natural for anyone to do. You just have to make sure that you have *all* of the files before you begin. Preflighting software helps a great deal with this.

Transition: I'm about to make the leap into preparing OEB-based eBooks, but before I do I'd like to introduce you to a key concept that most folks readily appreciate even if they weren't aware of it explicitly: from a content and design perspective books consist of 3 main elements…

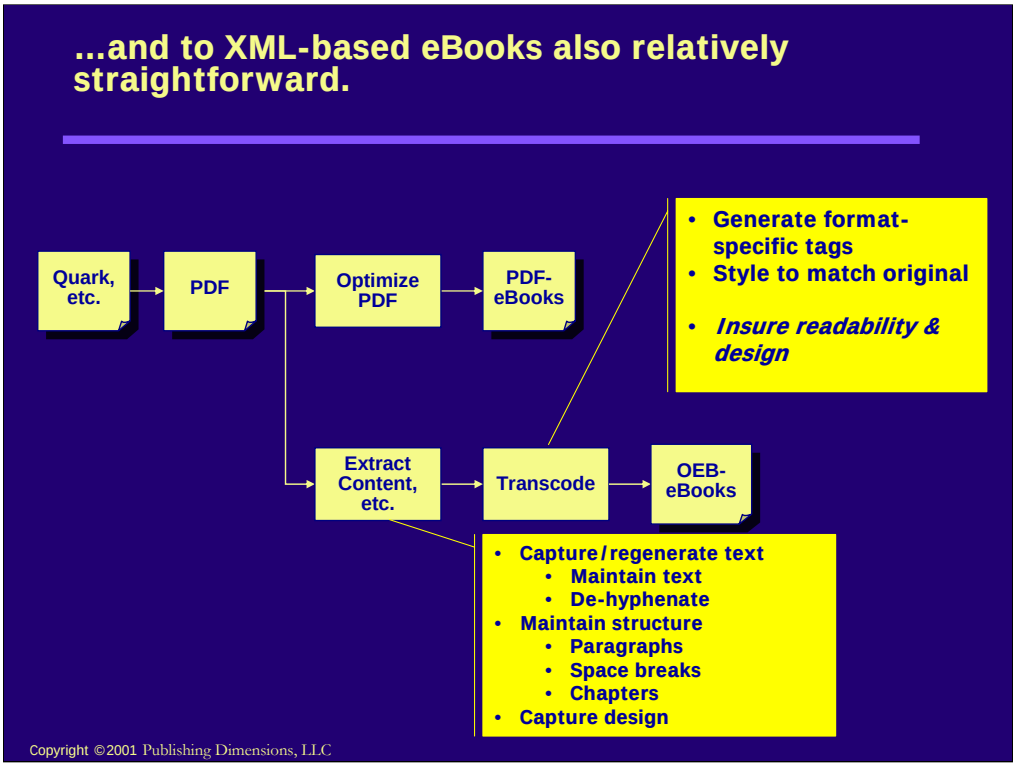


Content – semantic and syntactically tagged.

Structure – DTDs at the highest level or specific BOMs at the title level. The components that a particular title is made up of – which might not even exist until call for (configuration-on-the-fly)

All must be tied together with a common tag set – the structure of a book

Transition: Let's look in greater detail at content tagging

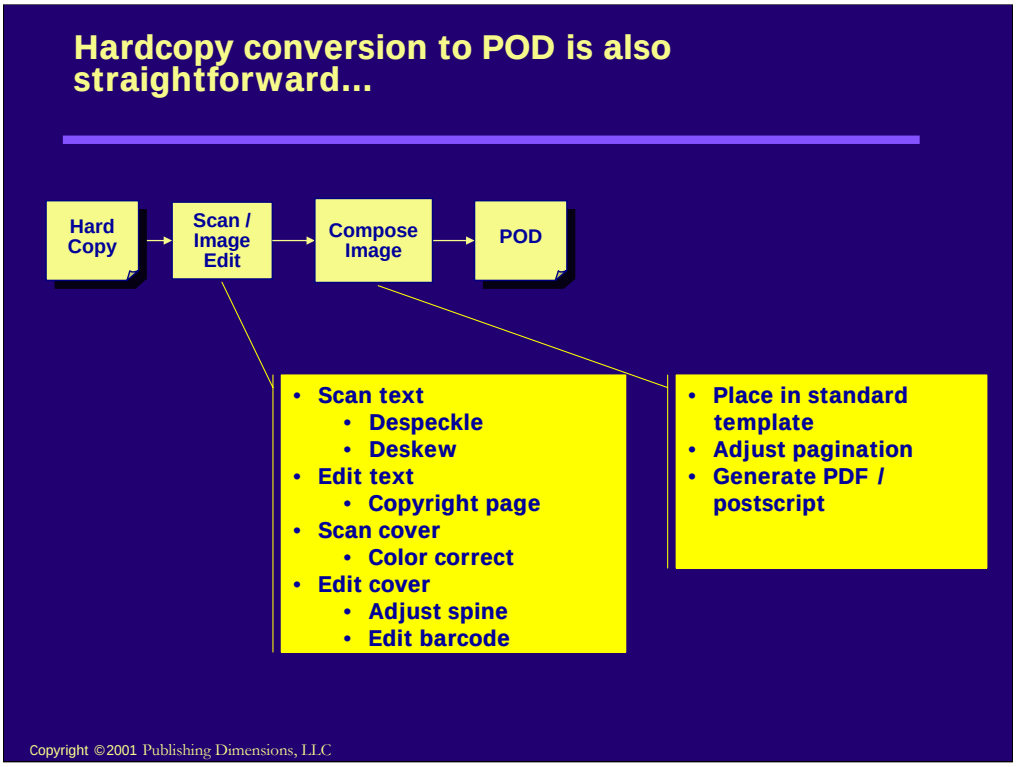


The main trick in converting PDFs to OEB-based eBooks is the extraction of the content, design and structure from the PDF. Remember you're extracting content, structure and design so as to match the final output as closely as possible to the original. There aren't very many good tools for this, although the accessibility plug-ins for Acrobat 5.0 are starting to get there. Texterity is making some progress here as are others out there, but this process still requires manual intervention. The main issues that I've seen are removing hyphens, dealing with strange fonts and undoing all of the tricks that designers used to make an attractive page. Also there are some books that defy creation of OEB representations just due to their sheer complexity – there are many textbooks like this. Coming out of this extraction process you'll end up with XML – usually OEB, and some sort of style sheet.

The next stage of the process is something I call "transcoding" mostly because I like the sound of the word. This is where you generate all of the different output formats and style them so that they look as close to the original as possible. These days if you store your intermediate format as some sort of XML, you'll likely use some sort of XSLT translations of the style sheets derived in the previous step. Coming out of this step you'll end up with the various formats that will display on the readers.

From all of this I'd like you to remember these two forks of the process – one to PDF eBooks and the other to OEB eBooks. BTW, the PDF editions can also be used for POD, although you would more than likely use different settings in distilling the archive formats.

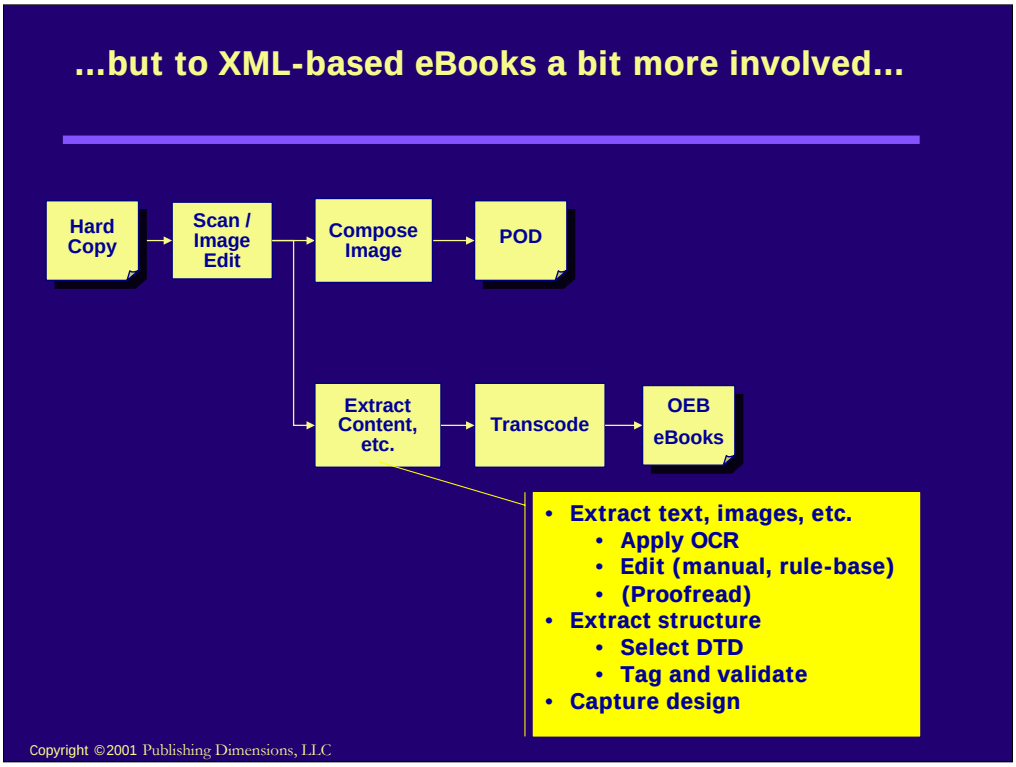
Transition: So where's the staging here? For that we need to look at repositories.



We start this process by debinding the book, going through a scanning and image editing process to take out skew and speckles and end up with clean page images. BTW, this isn't necessarily as easy as it sounds. I remember one vendor I was working with that had the tolerances on his "speckles" set a little to high. The program went right in and took out all of the periods and dots on the "i"s – it was most entertaining fixing that one!

The images are then slotted into a PDF template that positions them correctly for printing on a digital press such as at LSI, for example

Transition: The next step isn't so simple – getting to OEB-based eBooks

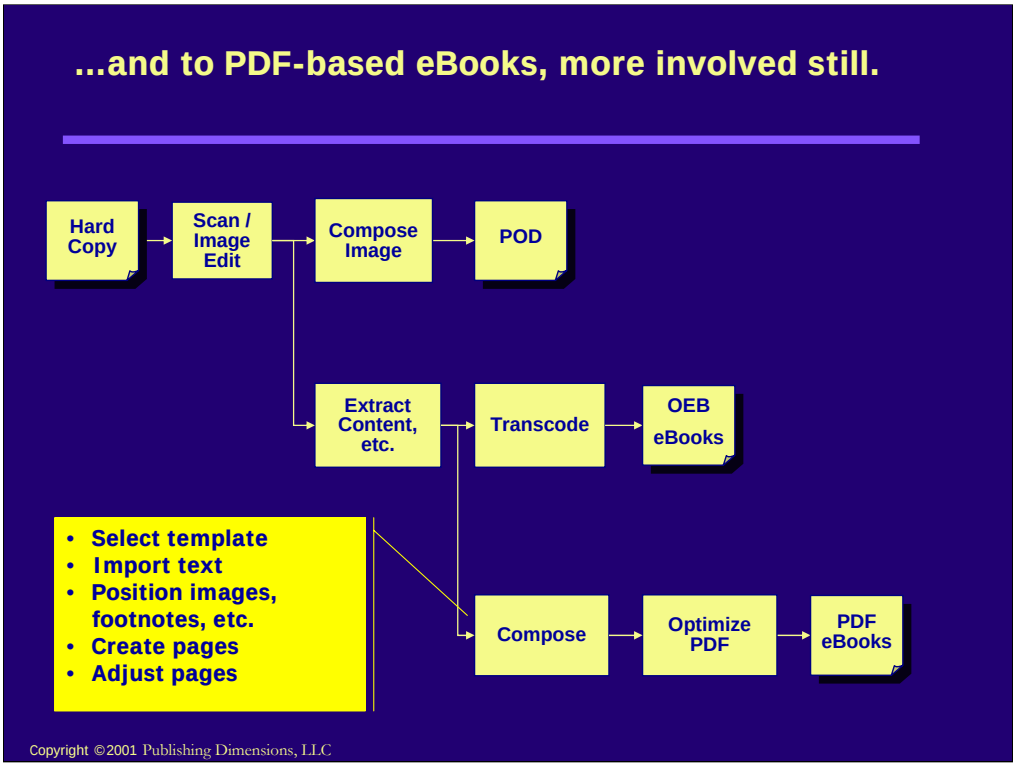


Here the Extraction process generally involves Optical Character Recognition (OCR) and a heavy dose of post-OCR editing. The content then has to be tagged into an intermediate format.

The big problem here is that there are very few clues as to the design. In a source PDF you get the names of the fonts to work with. Here you need a very experienced typographer or designer to tell you what the original fonts were.

Transcoding is the same as in the electronic editions – moving the title out to the various end-formats and making sure they are correctly styled.

Transition: The next step is to take the flowed text and regenerate nice-looking pages

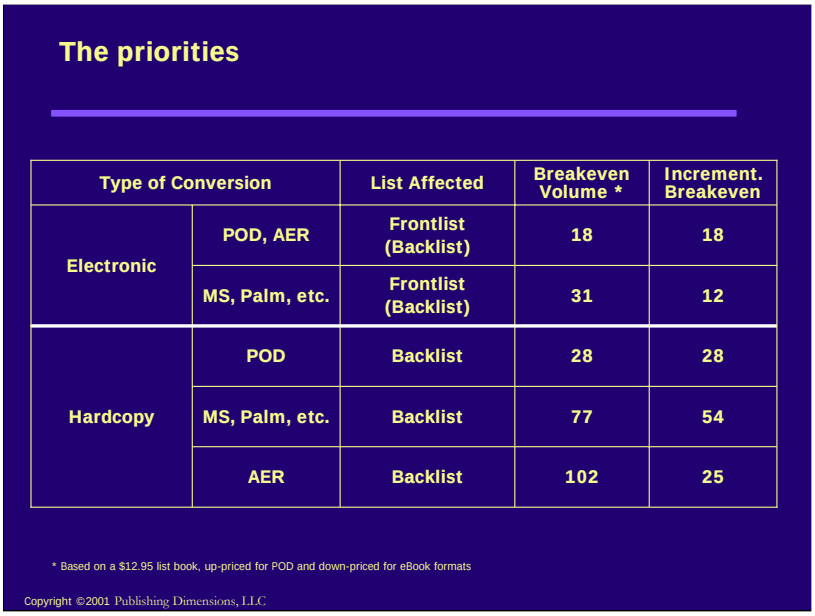


Here we're going through another page makeup process similar to what was done for the book originally. The difference is that the specs are coming from the original title, not a design.

I've heard of various experiments in round-tripping that have occurred, but this is still not easy to do, particularly if there is any kind of manual adjustment necessary to the pages.

We then go into the normal PDF optimization process to get AER editions.

Transition: Again there's an issue about where the repository sits



I've listed the 5 paths you can take up here as "type of conversion" You'll see that I've split them into rough families that align with the flow charts I went through earlier – the electronic family (from electronic files) and the hardcopy family (from bound books). I've also put up here what part of a publishers list is affected (frontlist or backlist), and most critically, what the breakeven volume is: the cost at which the gross margin from units sold pays for the cost of conversion. To get this I took conversion costs that are widely available in the industry and banged them up against a calculated Gross Margin per book. This GM included pricing with any required adjustments, discounts to the channel participants – I'm assuming retail distribution so there's a big hit there, cost of manufacturing if relevant, and cost of royalties to authors.

The final column is the incremental breakeven – that is if you take the previous step, what is the incremental units you'll have to sell to recover the extra cost of conversion. For example if we're in the electronic conversion family, and we've gone ahead and done the Acrobat eBook Reader format, how many more units would we have to sell to recover the cost of getting to the next version – say the Palm or Microsoft edition. The answer here is only 12 more units. For most books, I'd do this in a flash.

Let me go through what I see are the main conclusions here:

1. First notice that these breakevens aren't too high to begin with. The main point is if you have the book, you might as well get it into eBook and/or POD formats.
2. As you would expect for eBooks, it's much better to be working from electronic files.
3. As long as you're not competing with previously printed editions, it also seems like a slam-dunk to put all of your slow-moving titles into POD. The market is definitely there and you'll most likely sell far more than your breakeven

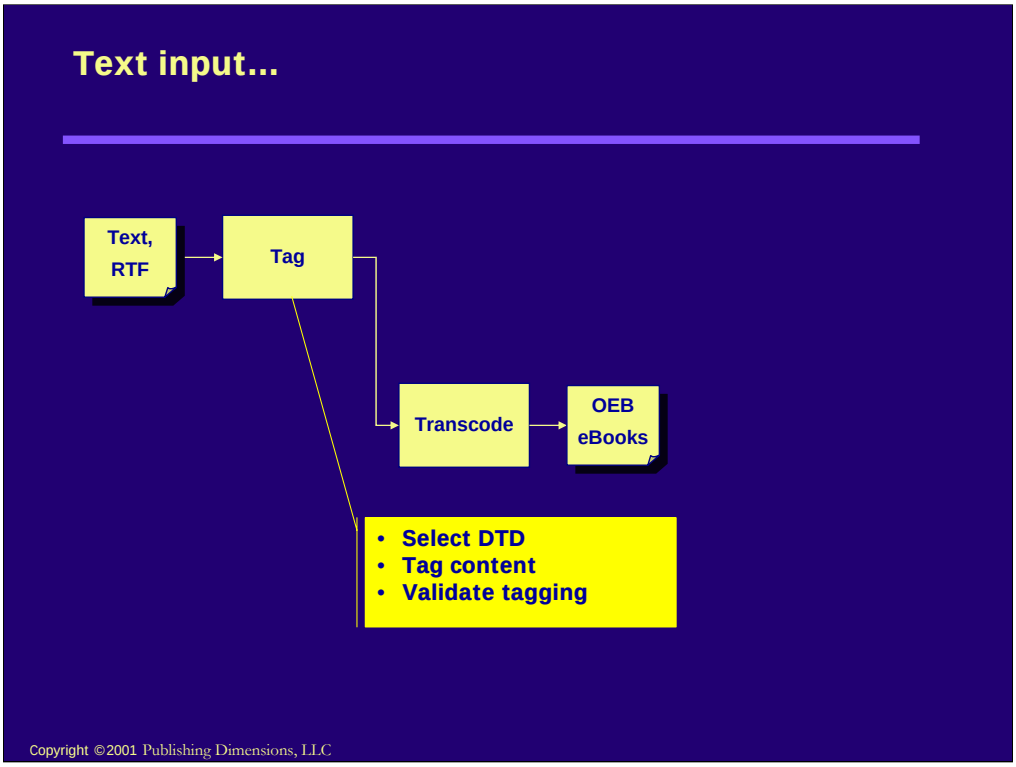
Transition: so that's the end of my material

Review:

1. I discussed some trends I'm seeing in publishing that are leading toward adoption of XML
2. I then talked about the use of XML in content and metadata and wrapped up with a discussion of areas to watch

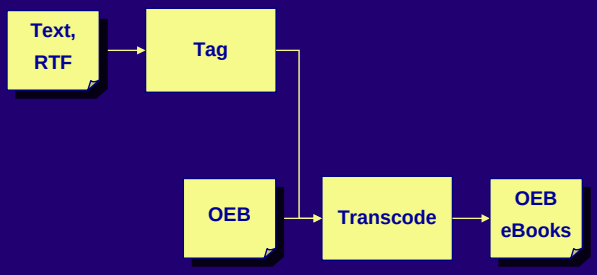
Importance: I think you'll agree that XML is becoming a major tool in cost control, in substantially increasing the breadth of offerings a publisher can provide and in substantially reducing time to market.

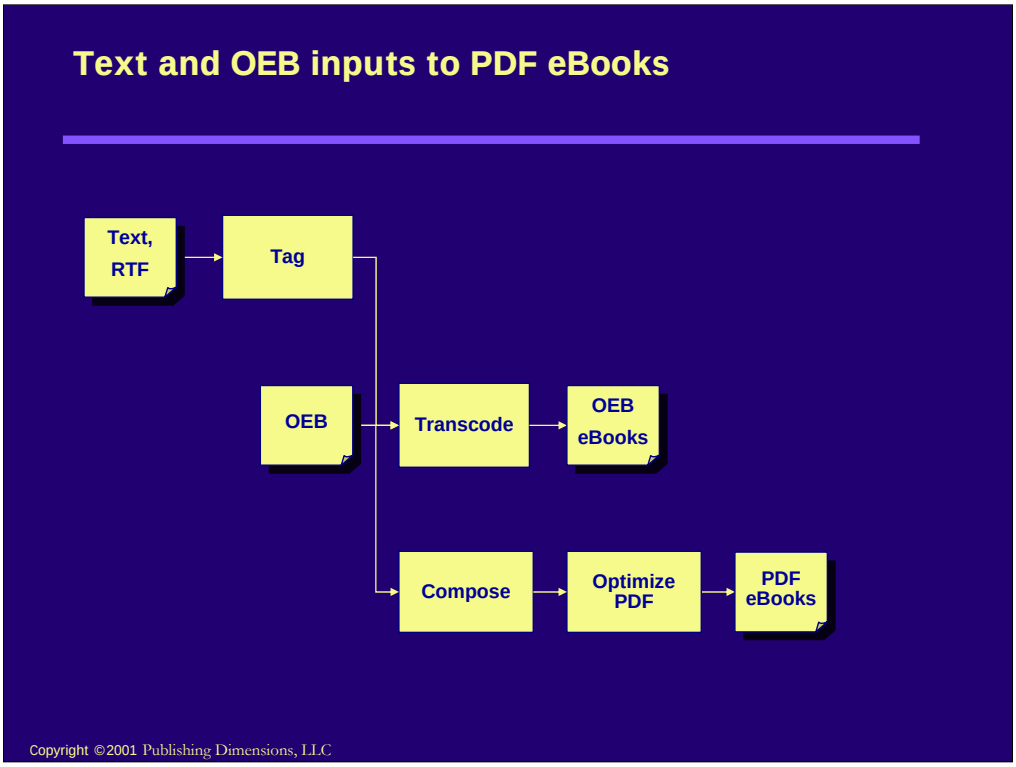
Transition: Katriel Reichman from LiveLinx will then go into live examples of how XML actually works in these applications



Experienced editors required here

OEB input…





Here we're going through another page makeup process similar to what was done for the book originally. The difference is that the specs are coming from the original title, not a design.

I've heard of various experiments in round-tripping that have occurred, but this is still not easy to do, particularly if there is any kind of manual adjustment necessary to the pages.

We then go into the normal PDF optimization process to get AER editions.

Transition: Again there's an issue about where the repository sits