

# Summary of Comments on XML\_AND\_PDF\_000912.ppt


---

Page: 2

---

Author: James King

Date: 9/12/2000 2:52:34 PM

 Thanks for checking out my presentation. I have annotated nearly all of the slides with notes like this one.

If you own Acrobat you can go to Tools->Annotations->Summarize Annotations and get a file of nothing but the annotations. Print it out and then look at the slides as you read the annotations.

Or you can just read them on the slides.

Jim King 9/12/00


---

# Page: 3

---

Author: James King

Date: 9/12/2000 2:53:21 PM

 This is the outline. The talk is broken roughly into two sections: the first an introduction to XML and the second some comments about where PDF is going related to XML.


---

# Page: 4

---

Author: James King

Date: 8/31/2000 11:10:42 PM

 HTML is the basis for the Web and as its name suggests it is a "markup language".


---

# Page: 5

---

Author: James King

Date: 8/31/2000 11:14:33 PM

 Here is a display of "plain text". This happens to be a plain text display of a memo that President Abraham Lincoln sent to Edward Everett the night before Lincoln delivered his now famous Gettysburg Address. (I'm having a little fun here imagining that Lincoln sent Everttt a courtesy e-mail to let him know what to expect.)

Edward Everett was the orator that spoke for nearly two hours just before Lincoln gave his less than three minute dedication. Lincoln was invited to give a short official dedication. He was invited to do this just a week or two before the event after the organizers found out he had decided to attend.

---

# Page: 7

---

Author: James King

Date: 8/31/2000 11:15:02 PM

 If we submit the plain text file to a web browser this is the result: plain text.


---

# Page: 8

---

Author: James King

Date: 9/12/2000 2:54:53 PM

 We would probably prefer to have the memo look like this page.

We do that by introducing styling directives to the plain text using "markup". Just as one might "mark up" a sheet of paper containing the plain text using a red pencil and indicating to a typesetter (person) what special treatment we want for the text to make it look nice, we have evolved systems where we mark up the text with more special text to indicate to the formatting and layout programs what treatment we would like to make it look nice.

We can even do things like introduce the image of the presidential seal.


---

# Page: 9

---

Author: James King

Date: 9/12/2000 2:55:39 PM

 Here is the exact marked up plain text that produced the preceding page. It is marked up in the HTML markup language. The original plain text is in black and the HTML markup is shown in red.

Note that on line 3 the presidential seal is introduced.

The spacing of the heading is controlled by using an HTML table and the markup for rows (<tr>) and for column data within each row (<td>).

All the special text styling is also done in this mark up indicating which font, size and color to use for the text that follows.


---

# Page: 13

---

Author: James King

Date: 8/31/2000 11:23:55 PM

 OK, now we have completed a quicky introduction to the notion of markup. Then what about XML, the Extensible Markup Language?

Despite its name, I claim it really shouldn't be called a markup language.


---

# Page: 14

---

Author: James King

Date: 8/31/2000 11:26:29 PM

 XML is more properly a set of "rules and tools" for creating individual markup languages. Many, many. Hundreds, possibly thousands of individual XML markup languages have already been invented.

Why would I want to follow some rules in defining my markup language. Well the benefits are listed.

---

# Page: 15

---

Author: James King

Date: 9/12/2000 2:59:33 PM

 First we are going to cover the basics of XML using my BusinessCard example. XML is really simple.

Next we will just tell you what DOM is without going into any details.

Then we will talk about some existing XML Languages that are standards.

And finally, in this section, we will talk about XHTML, the XML Language for HTML.


---

# Page: 16

---

Author: James King

Date: 9/12/2000 3:04:24 PM

 Here is an example of a particular XML markup language that I invented for this talk. It could be used to record the information found on a typical business card. The "plain text" is denoted in blue and the markup is in black.

It is important to understand that I can invent my own XML markup language for this purpose. Note that even if you can guess for what each element is intended you really cannot know the exact meaning of all the elements unless you ask me -- its my language. There are two additional things I will not be talking about that can be used to make more clear what the rules are for this particular XML markup language: a DTD or Data Type Definition and an XML schema.

But even with a DTD or an XML schema provided you may still not know everything you will need to know to properly understand and process one of my business cards unless you talk to me or I write down the details for you to read. We need to make sure people do this when they invent new XML markup languages.

I might write some software that would let me beam business cards between PDAs using my business card XML markup language.

---

# Page: 17

---

Author: James King

Date: 9/12/2000 3:03:53 PM




First of all XML is made up of "elements". The element "businesscard" is shown in red. Each element must start and end. Here we see that <businesscard> starts the element and the same notation with the introduction of a "/" provides the end element markup "</businesscard>".

---

Author: James King

Date: 9/12/2000 3:06:37 PM

 Here is another element "title".


---

# Page: 19

---

Author: James King

Date: 9/12/2000 3:09:36 PM

 Each element has "Contents" as show here where the "title" element has "Principal Scientist" as its content.

If an element has no content it can be represented with a begin/end shorthand as in:

`<title/>`

This notation is the same as if we used:

`<title></title>`


---

# Page: 20

---

Author: James King

Date: 9/12/2000 3:11:02 PM

 This shows that we can nest elements one within the other as the element "name" is nested within the element "company".

---

# Page: 21

---

Author: James King

Date: 9/12/2000 3:13:29 PM



This demonstrates that one can have the element repeated many times within an XML marked up document. Of course, we all knew that because we have used <p> elements to mark up paragraphs and we might need to do this hundreds of times in one document.

---

# Page: 22

---

Author: James King

Date: 9/12/2000 3:16:21 PM



The second key construct in XML markup besides the "element" is the named "attribute". As shown here the attributes follow the element name within the element begin markup and before the ">" that finishes the begin element markup.

Each attribute has its own "name" and "value" which are separated by an "=". The value must always be inclosed in quotation marks in XML. One of the rules.


---

# Page: 23

---

Author: James King


Date: 9/12/2000 3:18:20 PM

 Since XML elements form a nice nested list their structure can be represented as a graph (actually a directed acyclic graph or DIG). For example, here is the graph for a typical business card document.

---

Author: James King

Date: 9/12/2000 3:19:47 PM

 And we can also hang the "terminal" or "leaf" material off of this graph so that it is a convenient way to represent everything that is in an XML marked up document.


---

# Page: 25

---

Author: James King

Date: 9/12/2000 3:26:41 PM

 This graphical representation of an XML file is what the Document Object Model (DOM) provides in the computer memory. In addition, it supplies a API (application program interface) for use by programs and scripting languages to create and manipulate the graph structure and its terminal data.

The DOM is actually not tied to XML or HTML but just offers the APIs to create and manipulate the graph. It could be used for other things. It is very well suited to handling XML, however.

That is all that we are going to talk about the DOM but it is a key tool for processing XML files especially within a web browser. Most web browsers take a two step approach to processing HTML and XML: they first build a DOM tree and then walk the tree to present the material on the computer screen. This provides a means that a scripting program can animate material or modify it; it changes the DOM and then asks the browser to redisplay the modified material.


---

# Page: 26

---

Author: James King

Date: 9/12/2000 3:29:34 PM

 OK. Another check on our progress. So we have covered the basic two features of XML and at least introduced the concept of the DOM. This is really all there is to basic XML.

Next we will talk a little about the XML markup languages that have already been standardized by official standards organizations and others.

Then we will finish up the XML markup language part of the talk by noting the relationship between XML and HTML.

---

# Page: 27

---

Author: James King

Date: 9/12/2000 3:30:04 PM



Here are some specific XML markup languages that have been defined by official standards bodies or consortia. Again, each is its own independent markup language defined using the rules for XML. Each has additional documentation explaining the markup and the meaning for each element defined.


Notice the wide variety of things that the markup languages are used for.

If you really want to be overwhelmed check out the "XML Catalog" at the website noted. Click on the link -- it is live.

---

Author: James King

Date: 9/12/2000 3:31:48 PM

 OK so much for standards. Now on to HTML versus XML.


---

# Page: 29

---

Author: James King

Date: 9/12/2000 3:37:41 PM

 An important but minor variant of HTML has been defined called "XHTML". It is a version of HTML that follows all of the XML rules and therefore is one of the many, many XML markup languages. The original HTML didn't have as strict rules as XML has so HTML isn't strictly an XML markup language. For example, you can end one paragraph in HTML by just starting another one. This is not allowed in an XML markup language. Each element must have a start and an end. It is not allowed to leave off the end markup even if it is very clear.

XML also insists that all attribute values be enclosed in quotes. HTML doesn't require that. So the birth of XHTML; just like HTML but a legitimate XML markup language.

Note that this makes it really clear that XML is a set of rules and tools whereas HTML and XHTML are particular markup languages. XHTML being an XML markup language--one of many, many.


---

# Page: 30

---

Author: James King

Date: 9/12/2000 3:04:59 PM

 Big point of talk #1. It will generally be much more interesting to talk about a particular markup language designed for a particular purpose than it is to talk about the rules and tools provided for making XML markup languages.

In fact, I would go so far as to say you are not allowed to say "XML" unqualified anymore. You always have to talk about XML for something. You can see the examples of "XML for" statements that have a lot more meaning than saying "XML".

An analogy: many spoken languages use the "Roman alphabet". It isn't very interesting to talk about the Roman alphabet but there are lots of people that might want to discuss poetry written in the German Language.

There are at least two attributes that we can identify for a given markup language: what is it for (e.g., business cards) and what form or for what purpose was it invented. Is it for preserving the structure of the data, or is it to describe a flowable presentation markup like the HTML Gettysburg Address we saw earlier. Or it might be a final form presentation markup as provided by the SVG (Scalable Vector Graphics) XML markup language.


---

# Page: 31

---

Author: James King

Date: 9/12/2000 3:44:04 PM

 OK. Now to a section talking about why we have markup languages and how they are processed. We are especially interested in XML markup languages.

First we distinguish documents (data) marked up as data with a structure to it. The businesscard markup language comes to mind. These markup languages are primarily for storing and exchanging data. This is an extremely important and active area with respect to the World Wide Web right now. As companies are exchanging business data they are defining XML markup languages in which to do it.

There is also the area where we are marking up documents (data) so that it can be presented looking better on a computer screen or printed on paper. Our Gettysburg Address E-Mail example that we started the talk with is an example where we made it look quite nice by marking it up with "presentation" markup.


---

# Page: 32

---

Author: James King

Date: 9/12/2000 3:49:36 PM

 XML markup languages all share some key properties that make them great for storing and exchanging structured data.

First you can break your data down into "elements" thus distinguishing parts of the data as distinct. You can give these element types names and each part of the document that becomes an element can also have named attributes associated with it.

The nesting of elements provides a hierarchical data structure as we showed with the XML graph -- a very powerful concept we use all the time to organize our data.

And last, but not talked about here, is the idea that more complex data relationships can be show by explicit linking notations. An element might have a linking attribute that references other material via a URL or some other reference mechanism. Some markup languages use the attributes of ID="xxx" and later a REFID="xxx" to refer back to the other element.

Powerful notation for storing and exchanging data with structure!


---

# Page: 33

---

Author: James King

Date: 9/12/2000 3:50:55 PM

 This slide is introduced primarily to acknowledge a very interesting discussion that we are not going to pursue any further. If we are using an XML markup language whose intent is to capture the structure of some text or data we could be using it just as a great means to represent the data and to store it or exchange it. (Like the business card.) The use of the XML markup language may stop there. This is a very common and powerful use of XML -- making structural markup languages just to be able to represent and exchange some information. We aren't going to talk about this path any further today.

On the other hand, we might invent a structural XML markup language to hold and exchange data AND we might also want to present that information to you on a screen or on paper. This we will discuss further because it brings us to some very interesting things and leads back to the discussion we promised about the relationship between XML and PDF.

---

# Page: 34

---

Author: James King

Date: 9/12/2000 3:55:36 PM



Presentation mark up all has the property that it has elements and attributes for either determining how the material should look in a presentation or it at least gives directives and hints as to how it should be processed for presentation. Things like determining a font or a font size, the color of an element, and spacing material across a view by making tables.


Further, I divide presentation markup languages into two more categories: for flowable material and for final form material. Flowable primarily means that the presentation markup still has the notion of a paragraph that should be flowed into multiple lines of text. It isn't multiple lines of text now but later it will be.

Final form has the primary property that everything about the material's presentation is fixed. The positions of each element is already determined, the color, the font, and the position on the screen or page.

---

Author: James King

Date: 9/12/2000 3:56:29 PM

 OK. We've talked a lot about markup languages and some about what they are used for. Now lets talk about the tools available for processing the markup languages into different forms.


---

# Page: 36

---

Author: James King

Date: 9/12/2000 4:00:28 PM

 This is a diagram that represents the flow of processing "structural markup" into "flowable presentation markup" then to "final presentation markup" and finally onto your screen or onto paper.

This fits the emerging Web processing being defined by the W3C (World Wide Web Consortium at <http://www.w3c.org>).

This also roughly fits the processing done by batch formatters that have been around for years.

There are also interactive formatters like InDesign that don't quite fit the diagram because the user is involved in making many of the formatting decisions and that is not shown on the diagram. Sometimes interactive formatters don't take a markup language input either. They often accept plain text and the user does the markup in an interactive fashion.

But this is a good diagram for a general discussion and it matches the W3C processing model quite well.

I also make a call for us to standardize our terminology, especially with respect to formatting and rendering. For example, I have always used the term "render" to mean taking final form material and turning it into the bit maps or other representation needed for a particular device. Rendering doesn't involve making decisions about where some element is to be placed or what color it should be. That has already been done.

Similarly, formatting is determining the position of elements, possibly choosing a suitable font and definitely flowing text into fixed width areas.


---

# Page: 37

---

Author: James King

Date: 9/12/2000 4:02:15 PM

 You may, at first, think that web browsers don't fit this picture, but they do. They generally combine the formatting and rendering into one step. They usually don't materialize their final presentation material in any markup language that they output for us to see.

Interestingly enough, though, they sometimes do accept final form presentation material from other processors (plugins).

---

# Page: 38

---

Author: James King

Date: 9/12/2000 4:07:40 PM



I have broken out the style and template processing because today quite a bit of that is being done in web servers not the browsers. It is being separated out.

There are two mechanisms evolving for the web and being defined by the W3C for filling in templates and for adding styling information to XML (HTML) data.

XSLT defines a language for specifying transformations and XSL adds to that a "formatting objects" architecture that can be used to determine exactly how output will look (styling). XSL is great for doing template processing where the XSL file (represented in XML is the template and a structured XML file (think business card) is the data to fill in the template with.

CSS (Cascading Style Sheets) is a style sheet notation for adding the presentation properties to XML and HTML files.

There are new developments in this area every day and they are very exciting and already provide some very powerful tools to use on a server or in a client.


---

# Page: 39

---

Author: James King

Date: 9/12/2000 4:13:19 PM

 In the previous diagrams we showed "arbitrary" structured XML entering at the upper left and being processed down through the flow. Well there is a very important problem to be solved when you allow the system to take any old structured XML markup language.

For example, my markup language for business cards. There is no existing system that understands my business cards since I just invented them. So how do they get processed.

One way is in the styling and template phase they get turned into HTML or XMHTML. How does the styling mechanism know how to handle a business card. The answer is that I tell it by making business card style sheets or XSL processing files for business cards.

So the system doesn't have to "know" about each and every markup language that someone might invent. If the inventor is willing to provide the styling and/or templating information to get that language turned into HTML (XHTML) then the rest of the system knows how to turn this into a presentation. Systems exist around this idea that use other presentation markup languages besides HTML (XHTML).

---

# Page: 40

---

Author: James King

Date: 9/12/2000 4:14:15 PM



One reason I created this diagram was to show all the different ways that XML markup languages can be used. For each of the items shown in red there is an XML markup language defined by the W3C.

An interesting additional thing to observe is that you can take the same "document", for instance the Lincoln E-Mail shown at the beginning of the talk and pass it through this process generating XML versions of it in at least three distinct XML markup languages. The marked up version of the Lincoln E-Mail that was shown on slides 9-12 was at the second step: flowable presentation markup or HTML (XHTML). I created this version by first creating a structural form of the Lincoln E-Mail by inventing another XML markup language for "memos". It contained no presentation markup but just the "To:" "From:" heading information and the unformatted paragraphs of the body of the memo. I then created XSL files that when processed by an XSLT processor created the HTML (XHTML) you have seen. I then presented the XHTML to a web browser to create the "pretty" output I showed you earlier on slide 8. By a similar process I created yet another XML markup language version, an SVG version.

We will show you the details of this next.

If you are really interested you can contact me and I can send you all three (four counting the .xsl ) files.

---

Author: James King

Date: 9/12/2000 4:14:55 PM

 We start with an XML language for structural markup at the upper left of this diagram.

---

# Page: 42

---

Author: James King

Date: 9/12/2000 4:18:10 PM



This is a version of the Gettysburg Address e-mail that you haven't seen yet but is the starting place for all the Gettysburg Address files I have shown you.

This is not only a structural XML markup language it is also defined to describe "memos". It has a "heading" element and a "body". The heading contains the "to:", "from:", etc. material typical of a memo header.


The body contains the paragraphs of text that is the memo itself.

Note that there are not presentation markup in this form of the file. Just structural markup denoting what things are not how they should be presented.

---

Author: James King

Date: 9/12/2000 4:21:00 PM

 You have already seen the flowable presentation form of the Gettysburg Address e-mail. I used that to start the talk. However, I didn't tell you that that version was generated automatically using XSLT by feeding in the structural version I just showed you together with a template for memos style/transform XSL file. I don't show the XSL file here. It contains directives for introducing the XHTML table used to space the memo properly and it introduces the Presidential seal.


---

# Page: 46

---

Author: James King

Date: 9/12/2000 4:22:11 PM

 This is the XHTML file you already saw at the beginning of the talk. Notice that it contains lots of presentation directives, it is an HTML file (actually XHTML) but it still has the notion of a paragraph which hasn't been "flowed" yet.

---

# Page: 50

---

Author: James King

Date: 9/12/2000 4:24:38 PM



The third form of the Gettysburg Address I will show you is an SVG version which is in final presentation form. It has the paragraphs broken into individual lines. Each object in this representation is completely specified as to color, placement, size, etc.

I generated this version semi-automatically using the structural verions and a different XSL style file. However, there is no simple way to do paragraph flow in XSLT so I did a fudge on that by hand.


---

# Page: 51

---

Author: James King

Date: 9/12/2000 4:25:47 PM

 SVG is an XML markup language for final form presentations. It has everything nailed down. Each text segment is positioned to some x and y position on the drawing surface, the fonts are all determined, their size, etc.


---

# Page: 52

---

Author: James King

Date: 9/12/2000 4:26:38 PM

 Notice here that each line of text is separated out as an element and positioned explicitly. There is no flowing possible or required.

This is final form.

---

# Page: 55

---

Author: James King

Date: 9/12/2000 4:29:47 PM



So we have shown you three files, each in a different XML markup language, one for structure, one for flowable presentation and one for final presentation. The same Gettysburg Address material is within each file but markup up in a completely different way.

The second file can be made automatically from the first with the assistance of a style/template file (not shown). The third wasn't created completely automatically but in the future ... .

So if you still insist on talking about XML as if it was something besides rules and tools, which one of these files is THE XML file for the Gettysburg Address?


---

# Page: 56

---

Author: James King

Date: 9/12/2000 4:30:13 PM

 OK! What about PDF? You are the Adobe guy. I expected you to spend all your time touting the virtues of PDF. You haven't said a word about it yet.


---

# Page: 57

---

Author: James King

Date: 9/12/2000 4:30:37 PM

 Well, I think PDF is much more well understood than XML. We need to understand each well before we can talk about comparing them or their usages.

PDF is simple. It is a final presentation form for a document. I wouldn't go quite so far to call it a markup language but it definately fits into the diagram at the spot indicated.

Interestingly enough, that is also the spot where SVG and PostScript fit.


---

# Page: 58

---

Author: James King

Date: 9/12/2000 3:18:45 PM

 This is a rather wordy slide that I will let speak for itself.

Well I guess there is one thing to emphasize. I can capture the output of any formatting/layout process into a PDF file. Including from an interactive formatter where a user might take hours to fine tune the output to be exactly as desired.

Generally the "batch" processing implied by the XML web flow doesn't allow this user interaction so we have to settle for establishing "styling rules" that are then obeyed. The technology hasn't yet reached the point where pages created in a batch mode via styling rules can match the precise results obtained by a human. In some cases, this difference matters a great deal.

---

# Page: 59

---

Author: James King

Date: 9/12/2000 4:33:05 PM



From the first day that Acrobat and PDF were released our users have been asking for features. Today the clamour is for one file that will do many jobs.

Once text is broken into lines or "flowed" and we reduce the widow size through which we are viewing the text we may reach the point where the lines cannot be completely held within the window and still be large enough to be read. One is willing to do vertical scrolling to read text but horizontal scrolling is just not workable. (For latin languages written left to right.)

So one has to change the lines into shorter ones that do fit within the window. We call this "reflowing" the text.


---

# Page: 60

---

Author: James King

Date: 9/12/2000 4:33:54 PM

 Reflow with using XML processing.

Well the web processing we envision saves nothing along the way but always processes the XML from structural markup to final screen bit maps. This is its primary virtue. One can introduce the window size to the formatter and have it layout the text with a suitable line width.

Reflow is simply a matter of redoing all the work again.

The down side is that all the formatting must be controlled by styling rules and the technology just hasn't gotten to the point where we can get the precise output we want in all cases, with all window sizes with fixed preestablished styling rules.

---

# Page: 61

---

Author: James King

Date: 9/12/2000 3:01:42 PM




PDF poses a little tougher problem because it is sitting at the end of the formatter. In order to make the PDF files as small as possible the PDF format was originally designed to represent the final form only. One does not need to know that some red text was a heading or a bullet item in order to show it at the correct place and in the correct font and color on the screen. So the structural information has been generally discarded at each step in deriving the PDF. It is lost and in some cases impossible to reconstruct exactly correctly.

So in order to enable reflow, Adobe has defined places within the PDF file to preserve the structural information. Now we are urging those applications and processes that create PDFs to, optionally, write the additional structural information into the PDF file.

---

Author: James King

Date: 9/12/2000 4:35:28 PM

 Given that the PDF file has enough of the original structural information we can now reprocess it or parts of it to reflow the text for easy reading.

We demonstrated this technology on a Palm OS device as the Seybold Conference in San Francisco in September 2000.


---

# Page: 63

---

Author: James King

Date: 9/12/2000 4:36:06 PM

 XML is not a markup language. I'll let the slide speak for itself.


---

# Page: 64

---

Author: James King

Date: 9/12/2000 4:38:30 PM

 PDF is a great final form representation of multipage documents. Everyone keeps asking for more flexibility but, of course, they don't want to give up any existing features for it.

Our latest effort to make PDF more useful is adding the structural information to the document and then using that for "reflow".

It can also be used to create a structural XML representation of the document.


---

# Page: 65

---

Author: James King

Date: 9/12/2000 3:39:14 PM

 Thanks for sticking with me. Hope you enjoyed going through this presentation. Believe it or not I enjoy preparing it for you.

Jim King -- 9/12/00

---