Mixed content myopia

XML IN PRACTICE --- 07/11/2002

Sean McGrath

Have you ever had a debate with someone only to find that they are missing an important piece of knowledge? I humbly submit that the concept of "mixed content" in XML is one such concept.

I have been involved in debates about XML processing techniques that seemed to be going around in circles. More often than not, the disagreement stemmed from a different conceptual model of XML processing and, more often than not, that difference revolved around the important concept of mixed content in XML. If one party to a debate sees it in their mind-map of XML and the other does not, communication problems are likely to ensue.

What follows is a debug trace of a conversation bug attributable to mixed content myopia:

A: You know what XML is?

B: Sure. XML is a way of adding tags to text to denote the text's meaning. Tags come in pairs -- one at the start and one at the end.

A: Okay, but what about the text that is not tagged?

B: What do you mean, "not tagged"? If you add tags, you tag up all the text; there is no text outside of the tagging. Adding tags allows you to retrieve the values of those tags (i.e., the text between the start-tag and the end-tag) using APIs.

A: So you think of tag names as field names then?

B: Yes. Furthermore, XML APIs should allow me to read the value of any particular field.

A: Ah, okay. I see where you are coming from. Oh dear! Is that the time? Must dash!

The fact is, not all text is guaranteed to be tightly tagged in XML. Text occurring in, so-called, "mixed content models" can sit outside of any direct tagging. Further, such freestanding text can be freely intermixed with tags.

Such mixed content models offer both advantages and disadvantages. Being a natural model for narrative text offers a big advantage. For example, the p element in XHTML can contain mixed content as shown in the following fragment:

<p>This is a para with a <a href="foo">link</a> in it</p>

The value of the a element is the text "link", but what is the value of the p element? Is it the text before the a start-tag? The text before and after the a element? Does it include the word "link", which is in the a sub-element?

We could remove this ambiguity by fully tagging the text like this:

<p><text>This is a para with a </text><a href="foo">link</a><text> in it</text></p>

The advantage of this style is that each element either contains a fragment of raw text or a collection of other elements, but never a

- mixture* of both. On the downside, i is more complex and ugly to tag by hand.

Now you may be inclined to think that some hairs are being split here. Let us move over from doc-land to data-land and disprove this thought.

In the following XML fragment, what is the value of the <InvoiceNumber> element?

<Invoice><InvoiceNumber>42</InvoiceNumber><InvoiceValue>$24.42</InvoiceValue></Invoice>

I think you will agree that the value of the InvoiceNumber element is pretty unambiguously "42". Surely, XML APIs would provide some way to retrieve this value in a single call to some programmer's library? Something like GetValue("InvoiceNumber") for example?

This is where mixed content really bites. The most common APIs -- SAX and DOM -- provide nothing like GetValue. The API cannot know whether or not InvoiceNumber supports mixed content. If it does, no simple "value" can be associated with it.

Understandably, database programmers stare blankly at explanations of mixed content and think "this is Neanderthal isn't it?". They are right, in part. Yes, the standard APIs not better supporting such a common use case for XML is absurd; and yes, various attempts have been made to get around it. My own stuttering attempt was RAX[1]. More recently, a common API for XML Pull Parsing initiative has been formed [2].

Supporting pull APIs cleanly is made more complicated by the fact that you cannot know in advance if an XML document will contain mixed content. It is true that if you are validating the document as you parse, you can detect the presence of mixed content in the content models. Without the external indication, the poor parser has no option but to anticipate the presence of mixed content.

One possible way out of it would be to allow XML instances to declare, up front, that they do not use mixed content. That way, the parser could be smart about the API it exposes to the programmer providing pull features where possible.

[1] http://www.xml.com/pub/a/2000/04/26/rax
[2] http://www.xmlpull.org/ [3] http://www.rpbourret.com/xml/XMLDataBinding.htm

Sean McGrath is CTO of Propylon. He is an internationally acknowledged authority on XML and related standards. He served as an invited expert to the W3C's Expert Group that defined XML in 1998. He is the author of three books on markup languages published by Prentice Hall. Visit his site at: http://seanmcgrath.blogspot.com.

Mail to a friend

Home Newsletters XML IN PRACTICE