XML • text as data, data as text

Paul Hoffman · LIS 531F · Spring 2006

Introduced in 1998 as a self-delimiting, conceptually simple text-based syntax for constructing markup languages, XML has been adopted enthusiastically and is now used in applications as wide-ranging as lexicography (Thompson, 2005), business transactions (Ogbuji, 2003), and bioinformatics (Cohn, 2000). XML, like SGML before it, is a happy marriage of text and data, and promises the best of both worlds: a human-readable, largely uncomplicated text-based syntax for producing markup languages capable of representing both text and data with equal ease, and well suited to the treatment of text itself *as* data. SGML and XML are the inheritors of a long line of text-based markup, and they and their descendants are expected to thrive for many years to come.

It appears that little thought, however, has been devoted to the nature of markup, and even less to fundamental questions of what the meshing of text and data *means.* This brief note will attempt to raise a few of these questions, if not to provide the answers.

Text is neither precisely a transcript of spoken language nor a recipe for the production of speech; but neither does it stand wholly on its own, independent of speech. Like spoken language, it is subject to change, imperfect yet with evolved conventions that mostly fill the essential communicative needs of readers and writers. XML, expressed in the form of text, fulfills a long-standing expectation that data should be readable.

Data [< L. *datum* '(that which is) given'] are things "given or granted, … known or assumed as fact, and made the basis of reasoning or calculation" (Murray, 1933).

The use of text to represent data is, of course, as old as writing itself – or older. In the ancient Near East and in Pre-Columbian Mesoamerica, writing (or 'proto-writing') was used to record ownership, calendrical, and accounting data (Daniels & Bright, 1996). More recently, mathematicians have used our familiarity with text in speaking of formulations of 'grammars' that produce 'strings'. Even Gödel's revolutionary recasting of mathematical proofs as data takes advantage of our conventional view of text as a sequence of characters (Hofstadter, 1979).

The middle of the 20th Century saw the emergence of a new promise: text *as* data. From the fabled memex (Bush, 1945) to the use of a common markup syntax for bibliographic data and text (Mashey, 1976), the desire for a scholarly textbase was a major impetus behind the genesis of SGML and its forebears. The emergence of text in electronic form made possible concordances of Shakespeare, pattern-matching expeditions into Finnegans Wake, and other literary analyses; descriptive markup to bring additional richness to text and text processing.

Markup can be seen in such diverse practices as the cartouches of Egyptian hieroglyphic inscriptions, the cantillation marks of sacred Hebrew texts – an early form of procedural markup! – and our own Roman punctuation conventions. The last of these developed first as aids to new readers; the student of grammar and rhetoric was assisted by *punctūs,* an advantage scorned by the ancient *vir eloquentissimus,* for whom a full understanding of a text was gained only after long study and practice (Parkes, 1993, p. 12). Similarly, it was once common for children learning to read English to be helped in their efforts – or, perhaps, hindered – by the addition of macrons and breves to vowels, as for instance **spĭt** versus **spīte**.

Echoes of this long history of markup in its wide variety of forms can be seen in the happy dark days of early e-mail and USENET articles, before the commercialization of the Internet, when simple markup conventions arose that sprinkled asterisks and underscores – and, eventually, that ubiquitous attitudinal marker, the smiley face – throughout otherwise unadorned ASCII text. These mundane conventions have recently seen a rebirth in the formatting syntaxes of today's wikis (Cunningham, 2001).

Technological changes have wrought revolutionary transformations in our interaction with text; the most obvious example is the advent of HTML, which brought hypertext – albeit in a simplistic form – into widespread use. But perhaps a more striking example is the birth of the typewriter, which simultaneously enabled the rapid production of uniform texts and undid the work of centuries of typographers. Later developments – the teletypewriter and, especially, the phototypesetter – would bring procedural markup back into the picture, as a stream of bits could include instructions to a machine to move the print head or switch to a different typeface – or to ring a bell to summon a human operator.

The mingling of text and data doesn't stop there, however. For example, in the Unicode standard, characters are endowed with properties that enable richer text handling in software (The Unicode Consortium, 2003). For example, the *General category* property of the Devanāgarī character ७ (U+096D) is **Number, decimal digit**; its *Decimal digit value* property has the value 7. Unlike properties assigned by means of markup, these properties are considered intrinsic to the character and do not depend on the context in which it occurs. The advanced text handling capabilities that the formalization of these properties enables is another argument against the notion that text and data are distinct.

Meaning is the focus of unending speculation on the nature of reality and the place of humanity in the universe. For XML, more mundane questions of meaning may also prove problematic; even such a simple matter as determining where the meaning in an XML document resides is not as easy as it might seem. For example, Sperberg-McQueen, Renear, and Huitfeldt (2001) ask what (if anything) is the difference in meaning between pairs of XML fragments such as the two presented here:

```
<i>Now</i> is the winter          <i>N</i><i>o</i><i>w</i> is
of our discontent.                the winter of our discontent.
```

The reader will naturally assume that the two are equivalent; the characters wrapped in `<i>` tags are to be rendered in italics, and it makes no difference whether they are wrapped as a whole or individually. But consider the next two XML fragments:

```
<li>one</li><li>two</li>          <li>onetwo</li>
```

It seems clear that the author of these fragments meant to distinguish between the two. How, then, are we to know which is intended?

This may seem like a purely academic exercise, but the answer to questions like this is important in practical terms as well. For example, an XML editor that allows for the direct manipulation of element content must know what to do when the words 'baz' through 'krong' are cut in the following example:

```
<foo><bar>baz</bar>qux<bar>krong</bar></foo>
```

Is the correct behavior to remove just the content of the two `<bar>` elements, to remove the elements themselves, or to protest that an error has occurred? Nothing in any DTD or schema currently has the answer to that question, which means that XML itself cannot be accurately termed a self-describing language.

Of course, it should be pointed out that our ‘modern’ construct of text as data is nothing new, either: numerology, after all, depends heavily on this view.

When considering text as data and data as text, it is best to acknowledge that they are long intertwined in custom, and to look for ways of taking advantage of their happy and fruitful association. Just as so-called Western ‘print culture,’ with its notions of textual integrity and the author as sole guarantor of his work, did not suddenly appear out of nowhere upon the advent of moveable type (Johns, 1998), neither did the current practice of markup spring fully formed from the forehead of Charles Goldfarb and his peers.

⁂

Bibliography

Bush, V. (1945). As we may think. *Atlantic Monthly, 176*(1): 101–108.

Cohn, J. D. (2000). XML and genomic data. *ACM SIGBIO newsletter, 20*(3): 22–24.

Cunningham, W., & Leuf, B. (2001). *The wiki way: quick collaboration on the Web.* Boston: Addison-Wesley.

Hofstadter, D. R. (1979). *Gödel, Escher, Bach: an eternal golden braid.* New York: Vintage Books.

Johns, A. (1998). *The nature of the book: print and knowledge in the making.* Chicago: University of Chicago Press,

Mashey, J. R., & Smith, D. W. (1976). Document tools and techniques. *Proceedings of the 2nd International Conference on Software Engineering,* 177–181.

Murray, J. A. H. (Ed.). (1933). *The Oxford English dictionary.* Oxford: Oxford University Press.

Ogbuji, U. (2003). *Thinking XML: Universal Business Language (UBL).* Retrieved May 9, 2006, from http://www-128.ibm.com/developerworks/xml/library/x-think16.html

Parkes, M. B. (1993). *Pause and effect: an introduction to the history of punctuation in the West.* Berkeley, CA: University of California Press.

Sperberg-McQueen, C. M., Renear, A., & Huitfeldt, C. (2001). Meaning and interpretation of markup. *Markup Languages: Theory & Practice, 2*(3), 215–234.

Thompson. L. (2005). Project news: Pasadena. *OED News,* December 2005. Retrieved May 9, 2006, from http://www.oed.com/newsletters/2005-12/project.html.

The Unicode Consortium. (2003). *The Unicode standard: version 4.0.* Boston: Addison-Wesley.

⁂

Colophon

The text was prepared in OmniGraffle Pro and set in Adobe Garamond and Myriad.