

Why *eos*?

William F. Hammond

The GELLMU *article* document type has an end-of-sentence mark *eos*, which is a defined-empty XML element, corresponding to the concept of sentence in languages such as English and French. But there is no provision for regarding a western sentence itself as an XML element. Why?

There are two reasons.

Sometimes one wants to begin a *display* in the middle of a sentence. Then it can happen that the display is the last part of the sentence. It is a formal rule of XML that if an element begins inside another element, then the second element must be closed before the first element is closed. Following this rule, when a display is the last part of a sentence, the display must be ended before the sentence is ended. As a consequence an XML processor must usually work very hard to place the sentence-ending punctuation mark correctly.

Is this just a technical XML issue? Not really.

The second reason for modeling an end-of-sentence mark but not a sentence is that some literary use of a language such as English does not actually resolve into clean sentence units even though end-of-sentence punctuation is used.

One could argue that when a sentence is used, it could be marked up with a *sentence* element¹. In that event it is unlikely that authors would want to be required to insert end-of-sentence marks explicitly. Moreover, there would be something of a dilemma for the XML processor if it happens to notice an item of CDATA at the end of a sentence that appears to be an end-of-sentence mark. There would still be the vexation caused by a display that ends a sentence. And would authors use the *sentence* element?

Will authors want to use the explicit *eos* rather than the simple CDATA punctuation mark ‘.’? If so, how is the sequence “.*<eos/>*” to be handled by a processor?

Authors are the end users, and authors need convenience. Reasonable convenience lies in the convention that began with the dawn of the mechanical typewriter:

A sentence is ended with a period followed either by a newline or by two or more blank spaces.

Handling this convention is not a reasonably efficient task for an XML processor. But it works very well with a L^AT_EX-like markup interface for XML, i.e., when there is pre-processing from L^AT_EX-like markup to XML markup.

¹The model would then likely permit each of *sentence* and *display* to contain the other