

**Contents:**

- [An example of what TEI can do](#)
- [Other advantages of TEI](#)
- [Tools and TEI](#)
- [Resources](#)
- [About the author](#)
- [Rate this article](#)

Related content:

- [reStructuredText](#)
- [Roundup of XML editors](#)

Subscriptions:

- [dW newsletters](#)
- [dW Subscription \(CDs and downloads\)](#)

XML Matters: TEI -- the Text Encoding Initiative

An XML dialect for archival and complex documents

Level: Introductory

[David Mertz, Ph.D. \(mertz@gnosis.cx\)](#)Encoder, Gnosis Software, Inc.
04 Sep 2003

Nowadays, XML is usually thought of as a markup technique utilized by programmers to encode computer-oriented data. Even DocBook and similar document-oriented DTDs focus on preparation of technical documentation. However, the real roots of XML are in the SGML community, which is largely composed of publishers, archivists, librarians, and scholars. In this installment, David looks at Text Encoding Initiative, an XML schema devoted to the markup of literary and linguistic texts. TEI allows useful abstractions of typographic features of source documents, but in a manner that enables effective searching, indexing, comparison, and print publication -- something not possible with publications archived as mere photographic images.

The Text Encoding Initiative (TEI) is a decade older than XML itself, and older than other common documentation encoding XML schemas like DocBook. Specifically, TEI was developed -- in initial SGML form -- in 1987, almost an eternity in Internet time. Despite its age, TEI works at a different level than any other markup format that I am aware of, and remains the best solution to a certain class of problems.

Basically, TEI aims to encode all the semantically significant aspects of literary texts, both old ones that predate XML technology (or indeed, computers in general) and newly created ones. Certainly the words themselves are the most important semantic feature of prose or poetical texts. But throughout the history of print -- or of writing in general -- other typographic features have been added to texts to encode subsidiary aspects of their meaning. The use of presentation elements -- such as various types of emphasis, indentation and margins, tables, pagination, line breaks (as in verse), graphics, and decorations -- has enhanced, elaborated, or modified the meanings of the words in books, essays, pamphlets, flyers, bills, poems, liturgicals, and all the other forms literary works take.

Moreover, mere typographic features sometimes require an interpretive effort to fully decipher. As a trivial example, many books use italics to mark both foreign words and to mark the titles of other books. The semantic aspect of italicization depends on the verbal context, but clearly authors usually use such marks with distinct intentions. TEI aims to allow the markup of texts in a way that distinguishes all such meaningful aspects.

TEI is not really just an XML schema, it is more like a whole family of schemas, related in their general goal but varying in details of the tags and attributes used. In part, these schemas differ in being supported by different DTDs (or RELAX NG schemas). For example, TEI-Lite is a greatly simplified form of TEI that aims to support "90% of the needs of 90% of the TEI user community" (according to the TEI Web site). And other specializations are available as well. But even apart from actual specializations or subsets of the full TEI tag set, most users will utilize only a few of the tags available in the TEI DTD they are using. Different documents demand different markup, and different projects allow differing degrees of granularity.

An example of what TEI can do

Project Gutenberg is an effort to provide free versions of literary and historical works to a general audience. Thousands of titles have been transcribed and verified by the project's contributors. According to the "History and Philosophy" page on the Project Gutenberg site (see [Resources](#)), the goal is to produce texts in *plain vanilla* ASCII. For Project Gutenberg publications, any kind of emphasis is represented by capitalization, and paragraphs are divided with blank lines. While readers can reconstruct many conventional features of Project Gutenberg texts, TEI aims to mark these features explicitly. However, TEI is likely to be harder to *read* unless rendered in a prettified form through some transformation tool. But simultaneously, TEI is much easier to process and analyze with automated tools.

For example, Project Gutenberg makes available Shakespeare's *King Lear*. A short portion of this delightful play is transcribed as:

Listing 1. Project Gutenberg version of King Lear

Kent.
 Now by Apollo, king,
 Thou swear'st thy gods in vain.

Lear.
 O vassal! miscreant!

[Laying his hand on his sword.]

Alb. and Corn.
 Dear sir, forbear!

Kent.
 Do;
 Kill thy physician, and the fee bestow
 Upon the foul disease. Revoke thy gift,
 Or, whilst I can vent clamour from my throat,
 I'll tell thee thou dost evil.

A great deal of implicit semantic content could be added, using TEI. For example:

Listing 2. TEI version of King Lear

```
<sp><speaker>Kent</speaker>
<p>Now by Apollo, king,<lb/>
Thou swear'st thy gods in vain.<lb/></p></sp>

<sp><speaker>Lear</speaker>
<p>O vassal! miscreant!<lb/></p></sp>

<p><stage>Laying his hand on his sword.</stage><p>

<sp><speaker>Alb. and Corn.</speaker>
<p>Dear sir, forbear!<lb/></p></sp>

<sp><speaker>Kent.</speaker>
<p>Do;<lb/>
Kill thy physician, and the fee bestow<lb/>
Upon the foul disease. Revoke thy gift,<lb/>
Or, whilst I can vent clamour from my throat,<lb/>
I'll tell thee thou dost evil.<lb/></p></sp>
```

This markup is the same as that suggested by David Seaman in the article on this topic (see [Resources](#)). However, this style is perhaps still not sufficiently semantically rich. The tag `<lb/>` indicates a line break, which is simply a typographic feature that might be rendered in print. This is similar to HTML's `
` element, DocBook's `<LiteralLayout>`, or LaTeX's `\newline`. But TEI can be more specific if you wish to consider the verse structure of Shakespeare. For example:

TEI King Lear with explicit meter

```
<sp><speaker>Kent.</speaker><lg>
<l part="Y">Do;</l>
<l part="N">Kill thy physician, and the fee bestow</l>
<l part="N">Upon the foul disease. Revoke thy gift,</l>
<l part="N">Or, whilst I can vent clamour from my throat,</l>
<l part="N">I'll tell thee thou dost evil.</l></lg></sp>
```

Here Kent's speech is described as a **line group** rather than simply as a paragraph. Moreover, each line is optionally qualified -- the first as *metrically incomplete*, the rest as *metrically complete*. Such qualification is optional, and other `part` attribute values exist.

The degree of descriptive specificity lets scholars answer literary questions by automated means. For example, "Which speakers in Shakespeare's plays tend to speak metrically incomplete lines (and how does that influence the intended perception of those characters)?" Working from a simple printed version or from a markup format -- either purely typographically-oriented like LaTeX or XSL-FO, or one at a coarse semantic level like DocBook or HTML (or plain vanilla ASCII) -- does nothing specifically to aid such research. TEI brings some automation to many areas of literary scholarship.

Additionally, from a document preparation perspective, you are free to utilize rich semantic marks -- or to ignore them -- as the publication requirements demand. As a somewhat simplistic example, think of those editions of the New Testament that mark all the speech directly attributed to Jesus in red ink. A TEI markup could simply indicate speakers, then such typographic issues could be decided as part of the print process; there would be no need for

something like an explicit `color="red"` attribute in the markup. Other works could be prepared using similar conventions for marking significant elements of the text.

Other advantages of TEI

Obviously, most writing is not meter and poetry. But at every level, TEI offers varying degrees of typographic and semantic markup options. Understand here that the emphasis in TEI's typographic markup is not primarily focused on how a text should be rendered in future publication, but rather on how it was rendered in the past. For example, philosophical scholars who study Kant's *Critique of Pure Reason* refer frequently to the "A" and the "B" versions -- that is, Kant made a number of significant conceptual changes between his first and second edition. This convention is important enough that most editions of the *Critique* contain marginal notes indicating A and B page ranges (the two versions are now frequently included in the same publication as separate sections). The marginal notes refer to where given paragraphs occurred in the original (German) revisions; generally, the modern editions -- especially translated ones -- have quite different pagination than these first editions. TEI is probably the only markup convention in widespread use that suffices to properly annotate the *Critique*.

At an inline markup level, TEI allows for both typographic and semantic markup elements. For simple typographic notations, the tag `<hi>` can be used with the optional `rend` attribute. For example, `<hi rend="italics">` indicates that a given word or phrase was or should be rendered in italics. But if it can be determined *why* a phrase was italicized (it is both unambiguous, and sufficient effort is available to analyze the text), you might choose to use a tag such as `<title>`, `<foreign>`, or `<emph>`, which more specifically describe the reason why the author or publisher italicized the phrase. In addition, with the text so marked, you might decide, for example, to underline rather than italicize titles in a later edition.

The examples I have given only touch on the markup capabilities in TEI. TEI probably has more markup available than any one person can remember all at once. Fortunately, as I mentioned, TEI is generally designed to be usefully subsetted for specific tasks. For a certain goal or project, the best strategy is to decide in advance which few TEI tags you want to use. Developers, writers, or archivists can learn such a small subset with only a modest effort.

Tools and TEI

In a general sense, any tool that can work with XML can work with TEI. DTDs are available for several TEI variations, as are XSLT stylesheets of various sorts. Naturally, customizations for working with TEI in Emacs, Framemaker, and MS-Word can be found at the TEI Web site. An XMetal customization is also downloadable.

An interesting online tool provided by the initiative lets you customize an XSLT stylesheet to produce just the HTML output you desire. A Web form lets you select a variety of options, then returns a stylesheet reflecting your customizations (see [Resources](#)).

A number of scripts and tools are available for conversion of TEI-formatted documents into documents that are closer to the final print output. In the main, these target either LaTeX or XSL-FO as an intermediate format. These are the usual command-line tool chains that text processing programmers are accustomed to.

One tool I have grown quite fond of is the Java-based XML editor, oXygen. I have reviewed this product in the past, and since then it has continued to get better. In addition to being one of the first XML editors to incorporate RELAX NG support, the newest version of oXygen now includes a nice set of TEI templates -- just select one, and oXygen creates a document skeleton (and assists you in validation and tag entry as you go along). But most impressive of all, the XSL-FO stylesheets that also come bundled *just work*. I was able to create a couple of nice looking PDFs out of my TEI tests without spending hours configuring tool chains and reading obscure how-tos.

Resources

- Participate in the [discussion forum](#) on this article. (You can also click **Discuss** at the top or bottom of the article to access the forum.)
- Visit the [TEI home page](#) for more information on the Text Encoding initiative. Within the site, you'll find a number of resources, including an interesting look at a [Bare Bones subset of elements](#).
- The [TEI software page](#) offers links to software you can use to create, manage, and process TEI documents in SGML or XML.
- Take this tutorial on [TEI Lite](#), which is slightly more complex than the Bare Bones TEI.
- Check out the extremely admirable [Project Gutenberg](#), which has brought literary history to readers, free of charge and in electronic form, since 1971. A large collection of public domain literary works are available there, encoded as simple ASCII "etexts." One such work is Shakespeare's [King Lear](#), which I use as an illustration in this column.
- Find out more about the [history and philosophy](#) of Project Gutenberg.
- Read David Seaman's helpful discussion of the [King Lear](#) example.
- Develop custom HTML outputs with the online [XSL TEI HTML stylesheet parameterization tool](#).
- Check out the Java-based [oXygen XML editor](#), which includes a nice set of TEI templates.
- Find more XML resources on the [developerWorks XML zone](#). You'll find all previous installments of David's XML Matters column at the [column](#).

[summary page](#).

- IBM's [DB2](#) database provides not only relational database storage, but also XML-related tools such as the [DB2 XML Extender](#) which provides a bridge between XML and relational systems. Visit the [DB2 Developer Domain](#) to learn more about DB2.
- Find out how you can become an [IBM Certified Developer in XML and related technologies](#).

About the author



David Mertz once led the desperate life of scholarship. David may be reached at mertz@gnosis.cx; his life pored over at <http://gnosis.cx/dW/>. Suggestions and recommendations on this, past, or future, columns are welcomed. Check out David's new book [Text Processing in Python](#).



Rate this article

This content was helpful to me:

Strongly disagree (1) Disagree (2) Neutral (3) Agree (4) Strongly agree (5)

Send us your comments or click [Discuss](#) to share your comments with others.