

INTRODUCTION TO THE TEXT ENCODING INITIATIVE (TEI)

What is it and why should I care?

Joey Takeda

July 07, 2020

Digital Humanities Innovation Lab, Research Commons

Simon Fraser University





HI!

Joey Takeda, User Interface Developer, the Digital Humanities Innovation Lab at
Simon Fraser University



THIS WORKSHOP

THIS WORKSHOP

Brief conceptual introduction to encoding, XML, and TEI

THIS WORKSHOP

Brief conceptual introduction to encoding, XML, and TEI

Encoding practice!

All materials can be found here:

<https://sfu.ca/~takeda/teiworkshop/>



TEXT ENCODING AND THE TEI



THE PROBLEM

THE PROBLEM


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)





THE SOLUTION?

THE SOLUTION?

 < Text Encoding Initiative >




HomeGuidelinesActivitiesToolsMembershipSupportAboutNews

P5 Guidelines — English

P5: Guidelines for Electronic Text Encoding and Interchange

Version 4.0.0. Last updated on 13th February 2020, revision ccd19b0ba

[\[English\]](#) [\[Deutsch\]](#) [\[Español\]](#) [\[Italiano\]](#) [\[Français\]](#) [\[日本語\]](#) [\[한국어\]](#) [\[中文\]](#)

Front Matter

- [Title](#)
 - i. [Releases of the TEI Guidelines](#)
 - ii. [Dedication](#)
 - iii. [Preface and Acknowledgments](#)
 - iv. [About These Guidelines](#)
 - v. [A Gentle Introduction to XML](#)
 - vi. [Languages and Character Sets](#)

Back Matter

- Appendix A [Model Classes](#)
- Appendix B [Attribute Classes](#)
- Appendix C [Elements](#)
- Appendix D [Attributes](#)
- Appendix E [Datatypes and Other Macros](#)
- Appendix F [Bibliography](#)
- Appendix G [Deprecations](#)
- Appendix H [Prefatory Notes](#)
- Appendix I [Colophon](#)

Text Body

- 1 [The TEI Infrastructure](#)
- 2 [The TEI Header](#)
- 3 [Elements Available in All TEI Documents](#)
- 4 [Default Text Structure](#)
- 5 [Characters, Glyphs, and Writing Modes](#)
- 6 [Verse](#)
- 7 [Performance Texts](#)
- 8 [Transcriptions of Speech](#)
- 9 [Dictionaries](#)
- 10 [Manuscript Description](#)
- 11 [Representation of Primary Sources](#)
- 12 [Critical Apparatus](#)
- 13 [Names, Dates, People, and Places](#)
- 14 [Tables, Formulae, Graphics and Notated Music](#)
- 15 [Language Corpora](#)
- 16 [Linking, Segmentation, and Alignment](#)
- 17 [Simple Analytic Mechanisms](#)
- 18 [Feature Structures](#)
- 19 [Graphs, Networks, and Trees](#)
- 20 [Non-hierarchical Structures](#)
- 21 [Certainty, Precision, and Responsibility](#)
- 22 [Documentation Elements](#)
- 23 [Using the TEI](#)

TEI sourcecode

- [Getting and Using the TEI Sources.](#)
- [TEI GitHub Repository](#)
- [Bug Reports, Feature Requests, etc.](#)

[\[English\]](#) [\[Deutsch\]](#) [\[Español\]](#) [\[Italiano\]](#) [\[Français\]](#) [\[日本語\]](#) [\[한국어\]](#) [\[中文\]](#)

TEI Consortium | [Feedback](#)



THE TEI

A set of guidelines for encoding text

THE TEI

A set of guidelines for encoding text

A non-profit organization

THE TEI

A set of guidelines for encoding text

A non-profit organization

A community or consortium of users

THE TEI

A set of guidelines for encoding text

A non-profit organization

A community or consortium of users

Website: <https://tei-c.org/>

THE TEI

Is a markup language written in XML

THE TEI

Is a markup language written in XML

Currently in its 5th major revision (P5 4.0.0)

THE TEI

Is a markup language written in XML

Currently in its 5th major revision (P5 4.0.0)

Offers a rich vocabulary and method to encode:

THE TEI

Is a markup language written in XML

Currently in its 5th major revision (P5 4.0.0)

Offers a rich vocabulary and method to encode:

- **Bibliographic and structural features:** page breaks, headers, footers, page numbers, line breaks, divisions, paragraphs, line groups, etc

THE TEI

Is a markup language written in XML

Currently in its 5th major revision (P5 4.0.0)

Offers a rich vocabulary and method to encode:

- **Bibliographic and structural features:** page breaks, headers, footers, page numbers, line breaks, divisions, paragraphs, line groups, etc
- **Interpretative features:** stage movement, emphasis, place names, proper names, dialogue direction, etc

THE TEI

Is a markup language written in XML

Currently in its 5th major revision (P5 4.0.0)

Offers a rich vocabulary and method to encode:

- **Bibliographic and structural features:** page breaks, headers, footers, page numbers, line breaks, divisions, paragraphs, line groups, etc
- **Interpretative features:** stage movement, emphasis, place names, proper names, dialogue direction, etc
- **Editorial apparatus:** hands, witnesses, collation, gaps, additions, deletions, etc

THE TEI

Is a markup language written in XML

Currently in its 5th major revision (P5 4.0.0)

Offers a rich vocabulary and method to encode:

- **Bibliographic and structural features:** page breaks, headers, footers, page numbers, line breaks, divisions, paragraphs, line groups, etc
- **Interpretative features:** stage movement, emphasis, place names, proper names, dialogue direction, etc
- **Editorial apparatus:** hands, witnesses, collation, gaps, additions, deletions, etc
- **Linguistic features:** morphemes, feature structures, orthographic form, etc

THE TEI

Is a markup language written in XML

Currently in its 5th major revision (P5 4.0.0)

Offers a rich vocabulary and method to encode:

- **Bibliographic and structural features:** page breaks, headers, footers, page numbers, line breaks, divisions, paragraphs, line groups, etc
- **Interpretative features:** stage movement, emphasis, place names, proper names, dialogue direction, etc
- **Editorial apparatus:** hands, witnesses, collation, gaps, additions, deletions, etc
- **Linguistic features:** morphemes, feature structures, orthographic form, etc
- **Spoken features:** incidents, pauses, shifts, "communicative phenomenon", etc

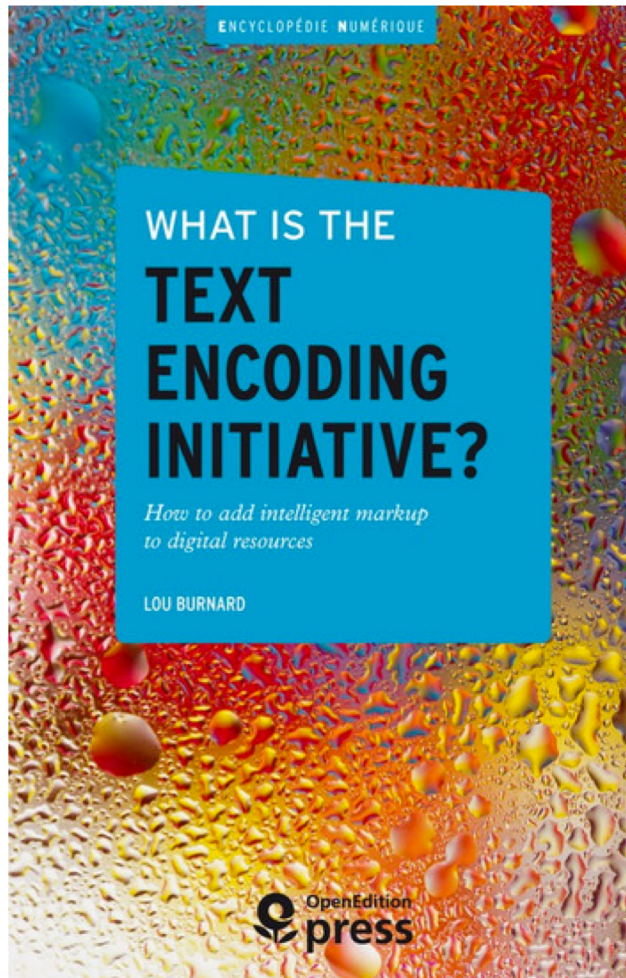
THE TEI

Is a markup language written in XML

Currently in its 5th major revision (P5 4.0.0)

Offers a rich vocabulary and method to encode:

- **Bibliographic and structural features:** page breaks, headers, footers, page numbers, line breaks, divisions, paragraphs, line groups, etc
- **Interpretative features:** stage movement, emphasis, place names, proper names, dialogue direction, etc
- **Editorial apparatus:** hands, witnesses, collation, gaps, additions, deletions, etc
- **Linguistic features:** morphemes, feature structures, orthographic form, etc
- **Spoken features:** incidents, pauses, shifts, "communicative phenomenon", etc
- **Metadata:** various classification schemes, provenance, manuscript description, etc
- ++++++



Within the noisy market place of the *Digital Humanities*, the TEI is a kind of senior member, an annoying parental figure for some, a benevolent one for others, something just too old-fashioned even to be considered for others. Yet, over the last decade, it has become increasingly clear that the TEI is part of what makes the digital humanities happen.

(Burnard, “Conclusion”, para. 1)

WHAT THE TEI IS **NOT**

WHAT THE TEI IS **NOT**

The TEI is **not** a language that describes how a text should be displayed online or in print. It should always concern the performative and expressive significance of the input, not the aesthetics of the output.

WHAT THE TEI IS **NOT**

The TEI is **not** a language that describes how a text should be displayed online or in print. It should always concern the performative and expressive significance of the input, not the aesthetics of the output.

The TEI is **not** a programming language; that is, encoding your texts in TEI does not automatically *do* anything to them

WHAT THE TEI IS **NOT**

The TEI is **not** a language that describes how a text should be displayed online or in print. It should always concern the performative and expressive significance of the input, not the aesthetics of the output.

The TEI is **not** a programming language; that is, encoding your texts in TEI does not automatically *do* anything to them

- Caveat: There are many, many tools for transforming TEI into other formats (Word documents, PDFs, and, of course, websites)

ENCODING, MARKUP, ET CETERA...

ENCODING, MARKUP, ET CETERA...

At its core, textual encoding is a way of identifying and differentiating bits of text from other bits of texts.

ENCODING, MARKUP, ET CETERA...

At its core, textual encoding is a way of identifying and differentiating bits of text from other bits of texts.

We do this all the time!

ENCODING, MARKUP, ET CETERA...

At its core, textual encoding is a way of identifying and differentiating bits of text from other bits of texts.

We do this all the time!

- Italics for *emphasis*

ENCODING, MARKUP, ET CETERA...

At its core, textual encoding is a way of identifying and differentiating bits of text from other bits of texts.

We do this all the time!

- Italics for *emphasis*
- Underlining for titles

ENCODING, MARKUP, ET CETERA...

At its core, textual encoding is a way of identifying and differentiating bits of text from other bits of texts.

We do this all the time!

- Italics for *emphasis*
- Underlining for titles
- Bold for **extra-emphasis**

ENCODING, MARKUP, ET CETERA...

At its core, textual encoding is a way of identifying and differentiating bits of text from other bits of texts.

We do this all the time!

- Italics for *emphasis*
- Underlining for titles
- Bold for **extra-emphasis**
- Quotation marks for “outside attribution” or “skepticism”

ENCODING, MARKUP, ET CETERA...

At its core, textual encoding is a way of identifying and differentiating bits of text from other bits of texts.

We do this all the time!

- Italics for *emphasis*
- Underlining for titles
- Bold for **extra-emphasis**
- Quotation marks for “outside attribution” or “skepticism”
- All capitals to YELL

ENCODING, MARKUP, ET CETERA...

At its core, textual encoding is a way of identifying and differentiating bits of text from other bits of texts.

We do this all the time!

- Italics for *emphasis*
- Underlining for titles
- Bold for **extra-emphasis**
- Quotation marks for “outside attribution” or “skepticism”
- All capitals to YELL
- +++

ENCODING, MARKUP, ET CETERA

But these are contextual and local

ENCODING, MARKUP, ET CETERA

But these are contextual and local

E.g. different types of punctuation for levels of quotation

ENCODING, MARKUP, ET CETERA

But these are contextual and local

E.g. different types of punctuation for levels of quotation

And they are subject to varying interpretations

- E.g. I think these quotation marks denote a term, but maybe the author is just being sarcastic...

WHY SHOULD WE ENCODE TEXTS?

WHY SHOULD WE ENCODE TEXTS?

Accessibility

WHY SHOULD WE ENCODE TEXTS?

Accessibility

Distribution

WHY SHOULD WE ENCODE TEXTS?

Accessibility

Distribution

Flexibility

WHY SHOULD WE ENCODE TEXTS?

Accessibility

Distribution

Flexibility

Interoperability

WHY SHOULD WE ENCODE TEXTS?

Accessibility

Distribution

Flexibility

Interoperability

Convertibility (i.e. from one format to another)

WHY SHOULD WE ENCODE TEXTS?

Accessibility

Distribution

Flexibility

Interoperability

Convertibility (i.e. from one format to another)

Analysis (Distant Reading, et cetera)

WHY SHOULD WE ENCODE TEXTS?

Accessibility

Distribution

Flexibility

Interoperability

Convertibility (i.e. from one format to another)

Analysis (Distant Reading, et cetera)

Answering existing (and asking new) research questions

WHY SHOULD WE ENCODE TEXTS?

Accessibility

Distribution

Flexibility

Interoperability

Convertibility (i.e. from one format to another)

Analysis (Distant Reading, et cetera)

Answering existing (and asking new) research questions



XML

XML

XML = e**X**tensible **M**arkup **L**anguage

XML

XML = e**X**tensible **M**arkup **L**anguage

XML is *not* a set language unto itself, but a grammar

XML

XML = e**X**tensible **M**arkup **L**anguage

XML is *not* a set language unto itself, but a grammar

XML is hierarchical

XML

XML = e**X**tensible **M**arkup **L**anguage

XML is *not* a set language unto itself, but a grammar

XML is hierarchical

XML is a tree-like structure

XML

XML = e**X**tensible **M**arkup **L**anguage

XML is *not* a set language unto itself, but a grammar

XML is hierarchical

XML is a tree-like structure

And is often described in genealogical terms

XML

XML = e**X**tensible **M**arkup **L**anguage

XML is *not* a set language unto itself, but a grammar

XML is hierarchical

XML is a tree-like structure

And is often described in genealogical terms

It is *not necessarily* a presentational format

- Some varieties of XML are (XHTML, SVG, et cetera)

XML IS EVERYWHERE

HTML (HyperText Markup Language: Every website)

KML (Keyhole Markup Language: Google Maps)

RDF (Resource Description Framework: Library catalogues)

SVG (Scalable Vector Graphics: Digital Images)

OOXML (Open Office XML: This presentation, word documents, et cetera)

XML

There is *nothing inherent about the function of XML*

XML

There is *nothing inherent about the function of XML*

It is purely a structure—a way of organizing

XML

There is *nothing inherent about the function of XML*

It is purely a structure—a way of organizing

Anyone can conceive of an XML dialect (e.g. it is *extensible*)

XML

Think of the hierarchy of the book:

XML

Think of the hierarchy of the book:

Book

XML

Think of the hierarchy of the book:

Book

- Chapters

XML

Think of the hierarchy of the book:

Book

- Chapters
 - Sections

XML

Think of the hierarchy of the book:

Book

- Chapters
 - Sections
 - Paragraphs

XML

Think of the hierarchy of the book:

Book

- Chapters
 - Sections
 - Paragraphs
 - Sentences

XML

Think of the hierarchy of the book:

Book

- Chapters
 - Sections
 - Paragraphs
 - Sentences
 - Words

XML

Think of the hierarchy of the book:

Book

- Chapters
 - Sections
 - Paragraphs
 - Sentences
 - Words
 - Letters



XML

XML

<book>

</book>

XML

```
<book>  
  <chapter>
```

```
    </chapter>  
</book>
```

XML

```
<book>  
  <chapter>  
    <section type="subsection">
```

```
      </section>  
    </chapter>  
</book>
```


XML

```
<book>
  <chapter>
    <section type="subsection">
      <paragraph>

          </paragraph>
    </section>
  </chapter>
</book>
```

XML

```
<book>
  <chapter>
    <section type="subsection">
      <paragraph>
        <sentence>

                                </sentence>
      </paragraph>
    </section>
  </chapter>
</book>
```

XML

```
<book>
  <chapter>
    <section type="subsection">
      <paragraph>
        <sentence>
          <word>

                                </word>
          </sentence>
        </paragraph>
      </section>
    </chapter>
  </book>
```

XML

```
<book>
  <chapter>
    <section type="subsection">
      <paragraph>
        <sentence>
          <word>
            <letter></letter>
          </word>
        </sentence>
      </paragraph>
    </section>
  </chapter>
</book>
```



XML EXPLAINED

XML EXPLAINED

The two pointy brackets is called an **element**

- E.g. `<book>` would be called the book element

XML EXPLAINED

The two pointy brackets is called an **element**

- E.g. `<book>` would be called the book element

All elements have **start** and **end tags**

- E.g. `<book>` is the start tag and `</book>` is the end tag

XML EXPLAINED

The two pointy brackets is called an **element**

- E.g. `<book>` would be called the book element

All elements have **start** and **end tags**

- E.g. `<book>` is the start tag and `</book>` is the end tag

XML EXPLAINED

The two pointy brackets is called an **element**

- E.g. `<book>` would be called the book element

All elements have **start** and **end tags**

- E.g. `<book>` is the start tag and `</book>` is the end tag

Elements can also have **attributes** and **each attribute must have a value**

- E.g. `<book type= "primary">` has a **type attribute with the value of primary**
- (Think of attributes as you would in everyday life; people don't have "height" or "age" without a value)



XML EXPLAINED

XML EXPLAINED

Elements **cannot** overlap

- `<sentence><word>Word1</word></sentence>` is right
- `<sentence><word>Word1</sentence></word>` is wrong

XML EXPLAINED

Elements **cannot** overlap

- `<sentence><word>Word1</word></sentence>` is right
- `<sentence><word>Word1</sentence></word>` is wrong

Elements **nest** and use genealogical terms

- I.e this bit of XML

```
<book>
```

```
  <chapter></chapter>
```

```
</book>
```

Can be described as “chapter is a child of book” OR “book is a parent of chapter”

XML EXPLAINED

Elements **cannot** overlap

- `<sentence><word>Word1</word></sentence>` is right
- `<sentence><word>Word1</sentence></word>` is wrong

Elements **nest** and use genealogical terms

- I.e this bit of XML

```
<book>
```

```
  <chapter></chapter>
```

```
</book>
```

Can be described as “chapter is a child of book” OR “book is a parent of chapter”

There is **always** a **root** element

- That is, there is always one element that encloses everything

WHAT TO ENCODE?

WHAT TO ENCODE?

Input \neq Output

WHAT TO ENCODE?

Input \neq Output

Encode what you care about and what you have time to encode

WHAT TO ENCODE?

Input \neq Output

Encode what you care about and what you have time to encode

If you don't encode it, you can't do much with it

RECALL: THE XML BOOK

```
<book>
  <chapter>
    <section type="subsection">
      <paragraph>
        <sentence>
          <word>
            <letter></letter>
          </word>
        </sentence>
      </paragraph>
    </section>
  </chapter>
</book>
```

THE PROBLEM

How else could it be written?

THE PROBLEM

How else could it be written?

```
<book>  
  <ch>  
    <para>  
      <w>  
        <c></c>  
      </w>  
    </para>  
  </ch>  
</book>
```



THE TEI SOLUTION

THE TEI SOLUTION

All texts must be called <text>

THE TEI SOLUTION

All texts must be called `<text>`

All divisions (whether they be chapters, sections, et cetera) must be called `<div>`

THE TEI SOLUTION

All texts must be called `<text>`

All divisions (whether they be chapters, sections, et cetera) must be called `<div>`

All paragraphs must be called `<p>`

THE TEI SOLUTION

All texts must be called `<text>`

All divisions (whether they be chapters, sections, et cetera) must be called `<div>`

All paragraphs must be called `<p>`

All words must be called `<w>`

THE TEI SOLUTION

All texts must be called <text>

All divisions (whether they be chapters, sections, et cetera) must be called <div>

All paragraphs must be called <p>

All words must be called <w>

+++

COMPONENTS OF A (BASIC) TEI FILE

COMPONENTS OF A (BASIC) TEI FILE

Root <TEI> element

COMPONENTS OF A (BASIC) TEI FILE

Root <TEI> element

A <teiHeader> that describes both the *file* and the *primary source* that you are transcribing (if applicable)

COMPONENTS OF A (BASIC) TEI FILE

Root <TEI> element

A <teiHeader> that describes both the *file* and the *primary source* that you are transcribing (if applicable)

A <text> that contains the text of the document

- Within text, you can have a <front>, <body>, or <back>

TEI IS FOR DATA AND METADATA

<TEI>

<teiHeader>

</teiHeader>

<text>

</text>

</TEI>

metadata

data

BASIC TEI FILE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <TEI xmlns="http://www.tei-c.org/ns/1.0">
3   <teiHeader>
4     <fileDesc>
5       <titleStmt>
6         <title>The Most Basic TEI File</title>
7       </titleStmt>
8       <publicationStmt>
9         <p>Not for publication, really.</p>
10      </publicationStmt>
11      <sourceDesc>
12        <p>No source, born digitally for demonstrative purposes.</p>
13      </sourceDesc>
14    </fileDesc>
15  </teiHeader>
16  <text>
17    <body>
18      <p>Hello, world!</p>
19    </body>
20  </text>
21 </TEI>
```




TEI

TEI

Note that the TEI is huge (569 elements)

TEI

Note that the TEI is huge (569 elements)

No one uses the entirety of the TEI tagset

TEI

Note that the TEI is huge (569 elements)

No one uses the entirety of the TEI tagset

Individual projects *customize* the TEI for their own needs, usually using a small subset of the overall tagset

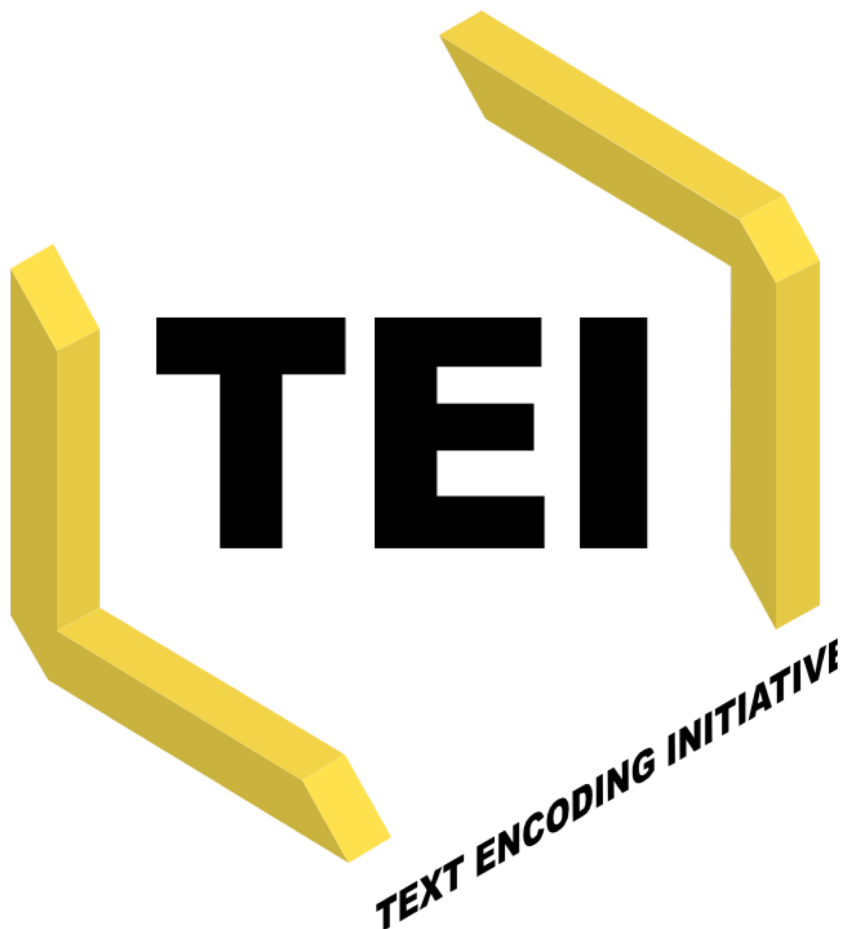
TEI

Note that the TEI is huge (569 elements)

No one uses the entirety of the TEI tagset

Individual projects *customize* the TEI for their own needs, usually using a small subset of the overall tagset

E.g. Drama projects will use the drama tagset (<sp> for speech, <speaker> for speaker, et cetera) and discard the linguistic/dictionary tagset (<entry> for dictionary entries, <m> for morpheme, etc).



LET'S ENCODE!

<http://www.sfu.ca/~takeda/teiworkshop/july07/>