The Text Encoding Initiative: Electronic Text Markup for Research

ABSTRACT

This paper describes the goals and work of the Text Encoding Initiative (TEI), an international cooperative project to develop and disseminate guidelines for the encoding and interchange of electronic text for research purposes. It begins by outlining some basic problems that arise in the attempt to represent textual material in computers and some problems that arise in the attempt to encourage the sharing and reuse of electronic textual resources. These problems provide the necessary background for a brief review of the origins and organization of the Text Encoding Initiative itself. Next, the paper describes the rationale for the decision of the TEI to use the Standard Generalized Markup Language (SGML) as the basis for its work. Finally, the work accomplished by the TEI is described in general terms, and some attempt is made to clarify what the project has and has not accomplished.

INTRODUCTION

This paper describes the goals and work of the Text Encoding Initiative (TEI), an international cooperative project to develop and disseminate guidelines for the encoding and interchange of electronic text for research purposes. In the simplest possible terms, the TEI is an attempt to find better ways to put texts into computers for the purposes of doing research that uses those texts.
The paper will first discuss some basic problems involved in that process, then some practical aspects of the reuse and reusability of textual resources. With the context thus clarified, the origins and organization of the TEI itself can then be described briefly, along with the reasons behind the decision of the TEI to use the Standard Generalized Markup Language (SGML) as the basis for its work. Finally, the work accomplished by the TEI can be described in general terms, and some attempt made to clarify what the project has and has not accomplished.

REPRESENTING TEXT ELECTRONICALLY

In the introductory paragraph of this paper, the TEI was described as an international cooperative effort to find better ways "to put texts into computers." The first problem encountered when one tries to set about this task is that, in a literal sense, it cannot be done. Texts cannot be placed inside computers—if only for the pedantic but simple reason that texts are abstract linguistic, literary, aesthetic, referential, historical, and cultural objects, while computers are physical objects controlled by complex electronic circuitry. Abstract objects cannot be "put into" physical objects. In this respect, text is on the same footing as numbers, which, being abstract mathematical objects, similarly elude any efforts to place them inside physical devices. The solution is the same in both cases: the best one can do is to make the physical object mimic the salient features of the abstract object and to manipulate this physical representation of the abstract object.

The value in this admittedly pedantic quibble is that it forces us to face squarely the critical fact that our problem is thus one of mimesis (or to put it into computational terms, one of finding a suitable representation for the data). Instead of a simple mechanical or quasi-mechanical process that can be carried out without reflection, the representation of texts in electronic form involves the same complications and limitations that inhere in any act of representation. Representations never reproduce all aspects of their objects with perfect fidelity; they invariably omit some aspect or other of the object represented and, by this omission, distort it. Designing a method for representing some object by means of some other object therefore ineluctably requires the designer not simply to decide what is salient and must be included but equally what is expendable and gets tossed off the sled in an emergency. It is no wonder, then, that systematic schemes for the representation of whole classes of objects reflect the biases, preconceptions, and preoccupations of their designers.
And yet for all their flaws, representations are absolutely essential to any intellectual work at all, because they are essential to understanding. Because they are selective reproductions of what is thought salient about some object, representations serve to reify our understanding of the object represented, and they allow us to test that understanding and compare it with different views of the object—themselves reified by different representations.

These issues are familiar, in a restricted form, to any computer programmer who has had to consider whether to represent a numeric quantity as a short integer, a long integer, or a real number at single or double precision; they are much less widely familiar when it comes to the representation of textual data in electronic form, even though textual data are intellectually much more complex and much less well defined than integers and real numbers—perhaps in part because text is less well defined.

If, as Niklaus Wirth has put it, "programs = algorithms + data structures," then a suitable method of representing textual data might be expected to represent a significant step forward in computational work with language and literature. Such a representation should make it easier to use computers to work with texts and thus contribute to the success of textual research and indirectly to the understanding of texts and of textual information.

If one asks oneself about the nature of a suitable representation for texts in electronic form, what it would mean to "represent a text" in a machine, one discovers a second advantage of the pedantic quibble with which this paper began. For, being forced to pose this question in terms of representations, one is equally forced to recognize that—since representations are typically utilitarian in character—the answer will inevitably be "it depends; suitable for what?" Before defining the qualities of a "suitable representation," one must specify what use is to be made of it. One is thus led to ask what it is that those interested in text in electronic form want to do with it.

A first simple answer is that we want to use it in the normal manner. Since it is text and we are readers, we will want to read it. Users will want to disseminate it to friends, colleagues, or the public across the network. As researchers, they will want to study it: literary scholars will want to study its themes, images, style, narrative structure, vocabulary, and diction; linguists will want to study its lexicon, morphology, parts of speech, syntax, or discourse structure. Textual critics will want to edit it, to study the variants in different manuscripts or early editions of the work, collate the various versions, and annotate it. Even those who work most intensively with computers will probably want to print the text out, nicely formatted, on paper. As time passes, the chances
are good that people will want to link the text to related material, be it other versions of the same text, commentary, graphics or illustrations, images of manuscripts, or yet other materials, either locally or in a network environment.

Equally important, we will want to reuse it. The costs of getting material into an acceptable electronic form are high enough to make reuse of data an important goal in virtually every computational field, from the natural sciences to the social sciences to the humanities. In the humanities, this fact is reflected in the increasing numbers of projects whose aim is to create generally usable bodies of electronic textual material intended for use by others; in computational linguistics, it is reflected in the growth of projects to develop standard reference corpora for use in all areas of natural-language processing, as well as in efforts to create "opportunistic corpora" gathering together as much textual material as can be obtained.¹

Third, because many of those interested in electronic text are researchers, it is a safe prediction that they will eventually want to do things with this electronic text that no one has yet invented or imagined. It is in the nature of research that not only the answers to the questions but frequently the questions themselves are not known at the outset of a project.

In other words, there is no satisfactory answer to the question of what we want to do with texts, once we put them into electronic form. In the long run, we want to do everything. This is not a wholly vacuous answer to the question; it does have the consequence that we want a representation that, as far as possible, does not constrain what we can do with the text. Anything we can do with the text, we would like to be able to do with the representation. It also serves to warn us that we should resist the temptation to design the electronic representation of text with any single application in mind. Since any given application for the electronic text is only one among many, there is not much point in designing characteristics into our data representation that make sense only in one application: a more general representation will make better sense in the long run, even if we must sacrifice some modest amount of short-term convenience or efficiency in a single application.

Paradoxically, experience seems to show that the best way to ensure that one can process the text in any way one wants is to ignore processing as far as possible and focus on saying what one thinks the text is. That is, one needs to find a declarative way of representing the text, not a procedural way. This involves adding a level of indirection to processing and so is sometimes disparaged as inefficient, but it's very important.
The basic problem of putting text into computers thus turns out to be that one must find a representation of the text that captures the essentials of the text and omits only the aspects one agrees to believe are negligible. In the practice of the forty-five years during which practitioners have been creating machine-readable texts for research purposes, one can identify some elements of a consensus regarding what is involved in such a representation. It is not enough to transcribe just the characters of the text; it is necessary to be able to include further information in the electronic text as well. This control information should ideally be readily distinguished from the text itself. Borrowing a term from traditional publishing, one can distinguish *markup* (the control information) from *content.* By means of explicit markup or otherwise, electronic representations of text must solve five problems.

First, they must find a method of representing the characters or symbols of the text. This is relatively simple in the case of the characters of the Latin alphabet, the Arabic numerals, and common punctuation marks; it is less simple for accented characters, special symbols, and scripts other than the Latin alphabet, because these are not well supported by common data-processing hardware or software. The situation is improving of late, with the development of ISO 10646 and Unicode, which provide a standard and very large repertoire of scripts and characters, but even with these standards, it will still be necessary to find ways to represent nonstandard symbols and characters (e.g., the special symbols of a personal shorthand invented by the writer of a manuscript or nonstandard characters omitted from ISO 10646 because they are nonstandard).

Second, they must represent, or choose to ignore, the overall logical and typographic structure of the text, including things like act and scene divisions and at least some phenomena like emphasis, quotation, bibliographic citation, and annotation. The history of typography offers persuasive evidence that these phenomena are important enough to thinking about texts that generations of scribes, authors, and typesetters have been forced to find print representations for them. Electronic representations of text would ignore the history of typography at their peril.

Third, the two-dimensional character of text in printed books and manuscripts must be reduced to a linear form in order to be represented in conventional computer file systems. This may involve changing the order of material (e.g., transcribing notes at their point of attachment), omitting material (e.g., running titles and page numbers, which are often omitted from electronic versions of texts), and finding methods of linking material that is physically separate but logically connected (e.g., endnotes).
Fourth, interpretive or analytic information is often explicitly represented, as in language corpora that tag each token with its part of speech. Such interpretive information may or may not be considered part of the text strictly speaking, but it is essential to certain kinds of serious work with the text. It is sometimes urged that creators of electronic texts eschew interpretation and limit themselves to the transcription of "the text itself." On this logic, for example, some would object to procedures like the provision of part-of-speech information in language corpora like the tagged Brown and Lancaster-Oslo-Bergen corpora on the grounds that it represents a subjective interpretation of the objective linguistic facts constituted by the wording of the texts.

As usually formulated, this objection to interpretation is intellectually problematic in itself, since no clear boundary can be drawn between interpretation and "the text itself." The "objective linguistic facts" about the wording of the text are themselves often the subject of hot disputes among textual critics, and even the reading of the characters in a manuscript (or in a printed book) can be controversial. That is, what constitutes objective fact for one reader may seem to another to involve illicit interpretation of the text. Those who create electronic text primarily for the use of others will of course do well to distinguish between information on which there is likely to be broad agreement and information more likely to be controversial, and to allow the user of the electronic text to disregard the controversial information in a systematic way. But it is impossible to distinguish consistently and firmly between controversial and noncontroversial information. And even if such a distinction were possible, it does not follow that electronic texts can or should be kept devoid of analytic or interpretive (i.e., controversial) information: as long as researchers use electronic texts in their work, they will find it convenient to record their interim or final results in the text, for further processing later on. Any general method of text encoding must therefore provide methods for recording such interpretations.

Finally, it is often useful to record certain auxiliary information about the text, even though it may not in any way be considered part of "the text itself." Control information identifying the author and title of the text, providing a bibliographic description of the source, identifying those responsible for the electronic version, and providing other useful information about the text, is commonly recorded in electronic texts or in accompanying documentation. A strong case for providing this information within the text itself can be made from the simple observation of how frequently electronic materials are found separated from the paper documentation that originally accompanied them. In language corpora, such ancillary control information may
often include characterizations of the text as a whole—e.g., demographic descriptions of the speakers in a corpus of spoken material or classification by subject matter and text type in corpora of written materials.

From the descriptions just given, it may be observed that in practice, the researchers who have thus far put texts into electronic form have been by and large more interested in texts per se than in the details of the pages on which the texts were written. The page is one representation of a text; the electronic transcription is another. The electronic version can of course represent the page, but it can also represent the text, without the intermediary of the page. For purposes of research with texts, what are needed are text description languages, not page description languages, and not just images of pages.

In emphasizing the text over the page this way, I follow the unspoken but unambiguous practice of standard practice in most textual work. New editions, even critical editions, very rarely preserve the pagination and lineation, let alone the typeface, leading, and gathering structure of earlier editions. This is only defensible if the text is not the same as the page. Often, students are given modern-spelling editions to read. This practice is defensible only if the text is not the same as the accidentals of the early printings or manuscripts.

Even though any scholar recognizes the potential importance of layout, typeface, etc., and is open to their overt or subliminal influence, still it is an unusual work of scholarship in language or literature (let alone the other disciplines that concern themselves with text) in which the argument hinges on typographic or bibliographic analysis. An obvious exception, of course, are works devoted to paleography, codicology, analytic bibliography, and the history of printing and binding. Practitioners in these fields will require methods of recording the details of the physical presentation of a text in a given edition or manuscript. Like other specialized information, however, this may not be of great utility to researchers in other fields.

RESOURCESHARING

Machine-readable texts have been in use for research for over forty-five years; this is about as long as computers have been commercially available. In general, computer-assisted projects of text analysis have historically followed a common pattern: first, the text to be analyzed is recorded in electronic form, and then the analysis itself is performed and the results published.

For at least thirty years, the observation has been made that when multiple projects work with the same text, the first step need not be repeated for each project. Once the machine-readable text is created,
it can be used for many different analyses without further encoding work. For thirty years, that is, there have been calls for machine-readable texts to be shared.

These calls for resource sharing, however, have been only moderately successful. Those concerned with encouraging sharing and reuse of resources might do well to ponder the reasons.

In the first place, some people don't want to share their texts. If one has gone to all the pain and trouble of creating an electronic text and is about to perform an analysis on it, one may well be reluctant to share it with others. These others may take it, perform their own analysis of it, and possibly even publish before the text's creator, receiving all the attendant glory. The creators of electronic texts may, however, wish to retain as much glory as possible for themselves, for use when they next come up for tenure, promotion, or a raise. The sharing of texts, however, confers much less glory than publication, and so creators of electronic texts have no incentive to share their texts and some incentive to retain them for private use.

It may be noticed that while in one light this line of argument is discouraging as to the prospects of achieving widespread reuse of resources, in another it is rather encouraging. The argument relies on the implicit claim that relative to the analysis the task of creating the electronic text is large and onerous. In other words, it really would save time and trouble for the research community overall if a way could be found to make it easier and more common to share electronic texts.

A second reason for the community's failure to achieve widespread text sharing is that when researchers do use each other's texts, they discover that they don't always understand them, because the methods used to encode the texts are so often idiosyncratic. This results in part from the newness of the medium. Faced with the task of representing a text in electronic form, without established conventions for the result, scholars find themselves in an Edenic position. Like Adam and Eve, the creator of an electronic text has the privilege of giving something a name, and having the name so given be the name of that thing. If one decrees, for example, that an asterisk is used to mark an italic word, and that a percent sign will precede and follow each personal name, and that a commercial at-sign is used to mark each place name, then that is what those things mean. The blankness of the slate gives to the encoder a kind of euphoric power, which is understandably slightly intoxicating. The result is that over the last forty years virtually every scholar who has created an electronic text has used the opportunity to wield that power and to invent a new language for encoding the text.

Electronic texts thus are, and have always been, in the position of humankind after the Tower of Babel. The result, predictably, has
been pretty much what the Yahweh of Genesis had in mind. The cooperation of the research community has been hindered and delayed by the needless misunderstandings and the pointless work of translating among different systems of signs, makework that would be unnecessary if there existed an accepted common language for use in the creation of electronic texts. Three distinct difficulties may be identified in the attempt by one researcher to use electronic texts created by someone else.

First, when one researcher (call her A) gets a text from another researcher (call him B), first of all, she may not understand what all the special marks in it mean. If B has invented a new language, a special system of signs, that is, for this specific text, then A may find that B's text contains signifiers that are opaque because A doesn't know their significance.

The second difficulty is that once A does understand B's signs, it may become clear that the signifieds of B's text don't tell her what she wants to know. It's good that A now understand that the at-sign means a place name, but if A is interested not in place names but rather in the use of the dative case (which B has not marked in the text), then B's text may not be as much use to A as she may have hoped before learning what all those special marks in the text meant.

The third difficulty is that, after swallowing her disappointment and beginning to add information to B's text, specifically by marking the occurrences of the dative case, A will all too frequently find

- that the markup language B used has no method of marking the dative case,
- that it also has no provision for graceful extension of its vocabulary, and thus
- that it does not scale up well.

THE TEXT ENCODING INITIATIVE

These three difficulties are not equally soluble, but they are all soluble at least in part. The TEI is an attempt to solve them, as far as possible.

The second is soluble only within very restricted bounds. Without violating the autonomy of the individual researcher, it is impossible to decree that we must all mark the dative case. Some of us, as it happens, are not interested in the dative case but concern ourselves instead with place names. It's hard enough to create texts suitable for our own purposes; we cannot hope to create texts suitable at the same time for everyone else's, too. Within limits, however, a tenuous consensus can be formed regarding some minimum set of textual features that everyone,
or almost everyone, regards as being of at least potential interest. No one should hope for too much from this consensus; the simple political fact is that very few features seem useful to absolutely everyone. Thus, I would not recommend to anyone that they should encode a text recording only the features that the universal consensus regards as useful. Almost no one would be happy with such a text; everyone regards other features as desirable, though we can reach no agreement as to what those other features are.

The first difficulty, that of understanding what it is the encoder is saying about a text, can be solved much more satisfactorily. The TEI will provide a large, thoroughly defined lexicon of signs (tags is the technical term) for use in marking up texts, and the published text of the TEI guidelines will suffice for virtually all the signifieds that workers with electronic text now record in their texts. By using this set of documented signs, one cannot guarantee that one will find the encoding work of others useful or interesting, but it can at least be made probable that secondary users of the text can understand what features the encoding of a text does and does not record.

Because such a vocabulary of tags must necessarily be rather large, almost no one will be interested in using every item in it. The first task of the encoder who uses TEI markup will therefore be to make a selection among the signs defined in the scheme and to begin making local policy decisions as to how those signs are to be used. The TEI provides, in the TEI header, a place to record those policy decisions, so that later users of the text can know what was done when the text was created.

The third difficulty, graceful extension and scale-up to more elaborate, information-rich versions of a text, the TEI handles in three ways.

First, the TEI itself is designed to be used both for rather sparse markup, which captures only a little information, and also for richer markup. That is, the TEI markup language itself scales up and down.

Second, the predefined vocabulary of the TEI includes a number of "built-in extensions," by means of which new varieties of known classes may be integrated into the markup scheme without any change to its formal definition at all. For example, many markup languages (TeX, LaTeX, Script, troff, Scribe) provide tags for marking enumerated lists, bulleted lists, and possibly one or more other styles of list. In general, however, one is limited to the varieties of list foreseen in the design of the system. One cannot add a new type of list to LaTeX without modifying LaTeX. The TEI defines one basic list element and provides a type attribute to allow different varieties of list (e.g., bulleted or enumerated) to be distinguished. Since the values of the type attribute
are not constrained, a new kind of list can be introduced simply by providing a suitable value for the *type* attribute.

Third, the definition of TEI conformance explicitly envisages the formal modification of the markup language itself, in cases where this is needed. The design and integration of such modifications do require a certain technical skill, though possibly less than is required to modify LaTeX or Scribe. But it is expected that, as with those systems, a local guru will usually be found who can help the user who needs help in changing the formal markup language.

The TEI thus builds a finite vocabulary but explicitly plans for its growth, both by means of formal modifications to the markup language and without such modifications, by means of built-in extensions. That is, the TEI explicitly recognizes that no finite vocabulary is complete.

The effort to solve the problems of interchange outlined above, by building such a scheme, began with a planning conference, held in Poughkeepsie, New York, at Vassar College in November of 1987. Thirty-one representatives of professional societies, research centers, text archives, and corpus projects met to discuss the desirability and feasibility of creating a single common scheme for encoding machine-readable texts. There was a clear consensus that such a scheme was both possible and desirable. Somewhat to the surprise of the organizers, this view was shared even by the participants responsible for several of the large existing archives of electronic text, many of which have thousands of dollars and tens of staff years invested in their own locally developed encoding schemes.

At the meeting, three organizations active in the application of computers to natural-language and textual material agreed to sponsor an effort to develop a new text encoding scheme, suitable for use both in local processing and as an interchange language between sites that preferred to use their own locally developed markup languages for local processing. These were the Association for Computers and the Humanities, which under the leadership of Nancy Ide had sponsored the planning conference, the Association for Literary and Linguistic Computing, and the Association for Computational Linguistics. Each of these associations named two delegates to a Steering Committee for the TEI, which began to meet almost immediately after the Planning Conference.

The Steering Committee, in turn, named the author as editor (later, Lou Burnard of Oxford University Computing Service was named as associate editor), with the responsibility of planning and coordinating the work of the project; sought and received funding from the National Endowment for the Humanities, the Andrew W. Mellon Foundation, the Commission of the European Communities (now the European
Union), and the Social Science and Humanities Research Council of Canada; and invited other professional societies to join in an Advisory Board. The Advisory Board met in February 1989, reviewing and approving the overall planning and design work done to that time. Following a plan for division of labor enunciated at the planning meeting, four working committees were appointed, with the task of addressing problems of

- text documentation (especially bibliographic control information and the like),
- text representation,
- text analysis and interpretation, and
- metalinguistic issues and syntax of the encoding scheme.

Of these, the first committee had the most clearly circumscribed area of responsibility, and the second and third had an essentially unbounded scope of activity. The slightly artificial distinction between representation and interpretation of a text was drawn for reasons of practical convenience. As a rule of thumb, the text representation committee was to be responsible for developing markup capable of recording the textual features signaled overtly (e.g., by italics, boldface, or special layout) by conventional printed books, while the committee on analysis and interpretation dealt with everything else that might be thought useful. The latter committee was instructed to concentrate its initial work on the problems of linguistic analysis, both because linguistic analysis seemed more successfully formalized than other textual disciplines and because linguistic understanding is a precondition of so many other areas of textual work.

The working committees met in 1989 and 1990, and the result of their labors was released in June of 1990 as TEI document TEI P1 ("public proposal no. 1"). In 300 letter-sized pages, this draft covered issues of characters and character-set documentation, defined a header for in-file bibliographic description of electronic texts and documentation of the encoding practices used in them, described SGML markup for a large set of features common to many text types and for the provision of analytic and interpretive information with particular reference to linguistic analysis, sketched SGML tag sets for corpora, literary texts, and dictionaries, and defined methods of extending the TEI tag sets.

After the publication of TEI P1, work immediately began on its extension and revision, and work groups were appointed to work on specialized topics such as character sets; textual criticism; hypertext and hypermedia; formulae, tables, figures, and graphics; language corpora; manuscripts and codicology; verse; drama and other performance texts; literary prose; linguistic description; spoken text; literary studies;
historical studies; printed dictionaries; machine lexica; and terminological data.

These work groups met over a period of two years, and the resulting draft, TEI P2, was issued chapter by chapter beginning in early 1992 and continuing through the end of 1993. At that time, all the published chapters were revised, several essential new chapters were added, and the resulting cumulative document was published in the first half of 1994 under the document number TEI P3. This version of the guidelines has grown from 300 pages to 1,300 pages, in part by the addition of an alphabetical reference list of SGML tags and in part by the addition of a great deal of new material. The following is the table of contents for TEI P3:

Part I: Introduction
1 About These Guidelines
2 A Gentle Introduction to SGML
3 Structure of the TEI Document Type Definition

Part II: Core Tags and General Rules
4 Characters and Character Sets
5 The TEI Header
6 Elements Available in All TEI Documents
7 Default Text Structure

Part III: Base Tag Sets
8 Prose
9 Verse
10 Drama
11 Transcriptions of Speech
12 Print Dictionaries
13 Terminological Databases

Part IV: Additional Tag Sets
14 Linking, Segmentation, and Alignment
15 Simple Analytic Mechanisms
16 Feature Structures
17 Certainty and Responsibility
18 Transcription of Primary Sources
19 Critical Apparatus
20 Names and Dates
21 Graphs, Networks, and Trees
22 Tables, Formulae, and Graphics
23 Language Corpora

Part V: Auxiliary Document Types
24 The Independent Header
25 Writing System Declaration
26 Feature System Declaration
27 Tag Set Documentation

Part VI: Technical Topics
28 Conformance
29 Modifying the TEI DTD
The design goals for the project were early formulated: The TEI encoding scheme should be

- sufficient for the needs of research;
- simple, clear, and concrete;
- usable without special software;
- rigorous and efficient to process;
- extensible; and
- conformant to existing and emerging standards.

These goals have not all been met in equal measure. The very size and subtlety required of the scheme by the first goal is partly at odds with the demand of the second goal that the scheme be simple, for example. In some measure, however, all of these goals have found some reflection in the final specification of the TEI encoding scheme:

- The list of topics given above, and the broad base of researchers who participated in the development of the guidelines, provide the best indication of the effort to ensure that the TEI guidelines would suffice to meet the needs of most researchers.
- In the interests of concreteness, the TEI formulated not general advice on the construction of SGML tag sets but a concrete TEI document type declaration (DTD), which can be used as is for the vast majority of research projects using electronic text.
- Because SGML is human-readable, software-independent, and requires no non-ASCII characters, TEI-encoded texts can in principle be used without special-purpose software, and interested projects can develop their own software to process TEI-encoded texts. Experience has shown, however, that work with TEI texts is materially aided by the use of SGML-aware software. This is particularly true of texts with complex encoding. To that extent, the third goal might plausibly be regarded as having been achieved only in part.
• Since the TEI scheme is formulated using SGML, it provides an explicit and rigorous document grammar and defines a tree-structured model of text (extended with pointers to allow the representation of directed graphs) that lends itself to efficient manipulation. To simplify the task of ad hoc software development, the TEI defines an “interchange format” that restricts the syntax of SGML to a manageable subset of the full syntax, which is thought by some to be marred by an excessive number of special cases and ad hoc rules.

• Extension of the TEI tag set is explicitly allowed in TEI-conformant texts—although this complicates the life of software developers materially and may make interchange more difficult and so is not actively recommended.

• The standards most relevant to text encoding are ISO 8879, which defines SGML, and the various character-set standards. SGML conformance is a condition of TEI conformance, but for pragmatic reasons, no single standard character set is mandated for TEI-encoded texts.

**TEI AND SGML**

As noted, the TEI uses SGML as the basis for its encoding scheme; this section describes the basis for that choice. First of all, SGML is nonproprietary, an international standard formulated by ISO (the International Organization for Standardization) and thus not within the control of any one software developer. This helps ensure the vendor- and platform-independence of SGML applications and of SGML-encoded data. With SGML, there is no user lock-in to specific systems; information is owned by the user, not by the propriety systems used to manipulate it. This is sufficiently important for industry to have led to wide adoption of SGML for strategic data. It is even more important for the research community, since computer systems commonly have lives measured in years, while major literary and linguistic research projects have lives measured in decades. Even for projects of shorter duration than the *Oxford English Dictionary* or its various counterparts in other languages, longevity is a major issue. Work in the textual disciplines may remain relevant and important for decades or centuries. When that work takes the form of electronic texts or work with such texts, it is important that the electronic forms of the texts remain usable for a much longer life span than any software has ever yet possessed.

Second, SGML provides a reasonably good model of text. Fundamentally, it allows text to be represented in a labeled tree structure,
with extensions to allow pointing and the creation of directed or undirected graph structures. A variety of mechanisms are available for handling information that does not fit well into a purely hierarchical model (discussed at length in one chapter of the guidelines). SGML is general, in contrast to markup languages like TeX or troff, which are focused on the production of printed output. It is extensible, in contrast to schemes like the Office Document Architecture (later renamed the Open Document Architecture), which do not allow for user extensions to the markup language. SGML-based markup languages are generally declarative, rather than procedural, and SGML encourages the use of analytic or descriptive, rather than appearance-oriented or presentational, markup. This helps achieve the reusability of SGML data.

THE TEI ENCODING SCHEME

The TEI encoding scheme is defined as an application of SGML, and its formal specification takes the form of an SGML “document type definition” or DTD. This specification is characterized by

- an emphasis on logical, rather than physical, structure of the text, on texts rather than on pages, for the reasons described above;
- the frequent application of Occam’s Razor—for example, in the provision of a single tag for lists, with an attribute to specify the type, rather than separate tags for ordered, bulleted, and simple lists;
- a modular architecture that groups tags into easily understood sets, which may be combined more or less freely for use with particular texts;
- the explicit provision of methods of adding new tags, and even new tag sets, to the encoding scheme, so as to ensure that the TEI markup language remains open to improvement and extension.

Particular attention has also been paid to ensuring that information of varying types can be included in the same document, and that documents can be gradually enriched by the addition of new information and analysis, without the new information getting in the way of the old. SGML software can readily ignore the markup not of interest to the user at any given moment, effectively filtering the document into a form suitable for the particular task in hand. It is possible using the TEI scheme, for example, to combine in a single document:

- orthographic transcription of the text;
- pointers to a digital or analogue recording of a speech signal or a videotape of an event;
- markup of proper nouns, dates, times, etc.;
- part-of-speech tagging;
• analysis of surface syntactic structure, including multiple analyses of ambiguous structures;
• analysis of the discourse structure;
• cross references to other material on the same topic;
• links to figures and graphics stored in any suitable notation (which need not be SGML).

A simple example may be used to show what the TEI scheme looks like in practice; most SGML-aware display software, however, will not show the tags to the user in this form, instead using font, type size, and layout guided by user-defined style sheets to signal the nature of the information being displayed.

A TEI-encoded version of Franklin Delano Roosevelt's first inaugural address, for example, might look like this:

```xml
<!DOCTYPE TEI.2 system 'tei2.dtd'>
<!ENTITY % TEI.prose 'INCLUDE'>
<!ENTITY wsd.en SYSTEM 'teien.wsd' SUBDOC>

<TEI.2>
<teiHeader>
<fileDesc>
<titleStmt>
<title>First Inaugural Address: An Electronic Version.</title>
<author>Franklin Delano Roosevelt.</author>
<respStmt><resp>tagged from the Project Gutenberg edition by</resp><name>C. M. Sperberg-McQueen</name></resp>
</titleStmt>
<publicationStmt>
<authority>C. M. Sperberg-McQueen</authority>
<pubPlace>Chicago</pubPlace>
<availability><p>This electronic text may be freely redistributed; it should not however be confused with the Project Gutenberg version of the same text, from which this version derives in part. The inaugural speech itself is in the public domain.</availability>
<date>1994</date></publicationStmt>
<bibl><title>"The only thing we have to fear...is fear itself." President Franklin Delano Roosevelt's First Inaugural Speech</title> [Originally delivered March 4th, 1933] ([Champaign, IL]: Project Gutenberg, 1994) [file fdr10.txt]</bibl>
</fileDesc>
</teiHeader>
<encodingDesc>
<sourceDesc>
<bibl><title>"The only thing we have to fear...is fear itself." President Franklin Delano Roosevelt's First Inaugural Speech</title> [Originally delivered March 4th, 1933] ([Champaign, IL]: Project Gutenberg, 1994) [file fdr10.txt]</bibl>
</sourceDesc>
</encodingDesc>
</TEI.2>
```

This tagged version of Roosevelt's inaugural was prepared as a demonstration of SGML tagging by C. M. Sperberg-McQueen. The untagged text from which it derives was produced by Project Gutenberg.
President Hoover, Mr. Chief Justice, my friends:

This is a day of national consecration, and I am certain that my fellow-Americans expect that on my induction into the Presidency I will address them with a candor and a decision which the present situation of our nation impels.

This is pre-eminently the time to speak the truth, the whole truth, frankly and boldly. Nor need we shrink from honestly facing conditions in our country today. This great nation will endure as it has endured, will revive and will prosper.

So first of all let me assert my firm belief that the only thing we have to fear is fear itself &mdash; nameless, unreasoning, unjustified terror which paralyzes needed efforts to convert retreat into advance.

In this dedication of a nation we humbly ask the blessing of God. May He protect each and every one of us! May He guide me in the days to come!
The document begins with an SGML document type declaration, indicating that the main DTD is found in a system file called "tei2.dtd"; on the second and third lines, entity declarations identify the identifiers "TEI.prose" and "wsd.en" with, respectively, the string "INCLUDE" and the system file "teien.wsd." The former indicates that the TEI base tag set for prose is to be included; the latter identifies an externally stored writing system declaration, which in this case documents the language (English) and character set used to encode the text. The string "]>" on the fourth line of the example ends the document type declaration.

The document instance itself begins on the fifth line. Each SGML element is delimited by a start-tag and an end-tag, themselves delimited by angle brackets or angle-bracket-slash and angle bracket. The "<TEI.2>" on line 5 and the "</TEI.2>" on the last line of the example show the beginning and end of the entire document instance. The root element, <TEI.2>, contains in turn two subelements: a TEI header, tagged <teiHeader>, and a <text>. The text itself contains merely a series of paragraphs, tagged <p>; the TEI header, on the other hand, has a fairly elaborate substructure used to document the electronic text, including its bibliographic source and the encoding practices used in creating it.

The allowable content (i.e., the syntax) and the semantics of the elements like <TEI.2>, <teiHeader>, and <p> are given by the TEI guidelines, as part of the predefined vocabulary of SGML elements provided by the TEI encoding scheme.

The TEI defines a single unified encoding scheme, which is scalable, allowing both very light text markup and extremely dense, information-rich markup. It provides explicit support for analysis of the text, without requiring adherence to any particular linguistic approach or other theory, and allowing the peaceful coexistence of many different types of analysis. Using standard SGML techniques, it makes possible the linkage of text to speech or other nontextual data at any desired level of granularity. With its wealth of flexible analytic mechanisms and its support for information filtering, the TEI encoding scheme provides a computationally tractable representation of rich text that has few serious competitors within or outside the SGML community. Above all, the work of the many volunteers on its work groups has ensured that the TEI defines a compendious inventory of textual phenomena.
of interest to researchers, for the description of the physical, formal, rhetorical, linguistic, and other aspects of the text.

CONCLUSION

By providing a common public vocabulary for text markup, we will have taken one major step toward making electronic texts as important and useful as they ought to be, but only one step. Other steps are still required.

First of all, we must as a community make a serious commitment to allowing reuse of our electronic texts. This will require either a massive upsurge in the incidence of altruism or much stronger conventions for the citation of electronic texts, and giving credit for the creation of electronic materials, both in bibliographic practice and at promotion, tenure, and salary time.

Second, we must cultivate a strict distinction between the format of our data and the software with which we manipulate it, because software is short-lived, but our texts are, or should be, long-lived. Our paper archives are full of documents 15 or 20 years old, or 150 to 200 years old, or even 1,500 or 2,000 years old. But I cannot think of a single piece of software I can run that was written even 100 years ago. To allow our texts to survive, we must separate them firmly from the evanescent software we use to work on them. SGML and other standards encourage such a distinction, but proprietary products typically obscure it. In some operating systems, every document is tied, at the operating system level, to a single application—precisely the wrong approach, from this point of view.

Third, we need to cultivate better, more intelligent software, with better understanding of the nature of text structures, in order to make the texts contained in our archives more useful in our work.

Finally, we need, if possible, to come to a richer consensus about the ways in which we encode texts. We should try to move beyond an agreement on syntax and achieve more unity on the specific features of text that are widely useful. Such a consensus will make the TEI less of merely syntactic convention and more of a real common language.

The TEI's contribution to the success of electronic textual research will, I hope, be that it provides us with a common language, to allow us to escape our post-Babel confusion. As the list just concluded makes clear, such a common language is not all we need. But as the Yahweh of Genesis says:

If as one people speaking the same language they have begun to do this, then nothing they plan to do will be impossible for them. (Gen. 11:6, New International Version)
NOTES

1 Among humanities projects, one might mention the Brown University Women Writers' Project, which is creating a corpus of women's writing in English from 1330 to 1830; the Nietzsche Nachlass project now at Dartmouth; the Leiden Armenian Database, collecting primarily medieval Armenian texts; the Global Jewish Databank at Bar Ilan, an outgrowth of the earlier Responsa Project, a collection of rabbinical responses to questions on points of Jewish law; and the Thesaurus Linguae Graecae at the University of California at Irvine. This is by no means an exhaustive list but indicates the breadth of current activity. Among corpus projects, the Brown and Lancaster-Oslo-Bergen corpora of the 1960s, and their various analogues in other languages, are now being succeeded by a new wave of larger projects, for example, the British National Corpus, which will encode 100,000,000 (one hundred million) words of written and spoken British English, and the Network of European Reference Corpora. The most prominent of what I am referring to as “opportunistic projects” may be the ACL Data Collection Initiative (DCI) and the European Corpus Initiative (ECI).

2 There are occasional efforts to argue that markup is not necessary and, indeed, is actively harmful. Perhaps the most widely known proponent of this view at the moment is Michael Hart of Project Gutenberg, which distributes ASCII-encoded public-domain texts by means of anonymous File Transfer Protocol (FTP) servers. Each Project Gutenberg text, however, appears to contain an extensive header, giving the text's version number, filename, and date, providing a contact address, appealing for funds, and including a lengthy legal disclaimer. This header provides metatextual information, which is not strictly part of the text being transcribed, and so by definition constitutes markup of the text. Thus, even those who resist the use of formal markup languages do recognize in practice the need for markup to provide meta-information. One drawback of providing such meta-information without a formal markup scheme is that there is no convenient method to recognize automatically the boundaries between the text and the meta-information or markup.

3 I take Father Roberto Busa's Index Thomisticus project, which began in 1948, as marking the first use of machine-readable text for research.

REFERENCE
