Journal of the Text Encoding Initiative

Issue 12 | July 2019 - May 2020

Selected Papers from the 2017 TEI Conference

What is TEI Conformance, and Why Should You Care?

Lou Burnard

Electronic version

URL: <http://journals.openedition.org/jtei/1777>

DOI: 10.4000/jtei.1777

ISSN: 2162-5603

Publisher

TEI Consortium

Electronic reference

Lou Burnard, « What is TEI Conformance, and Why Should You Care? », *Journal of the Text Encoding Initiative* [Online], Issue 12 | July 2019 - May 2020, Online since 06 December 2018, connection on 25 June 2020. URL : <http://journals.openedition.org/jtei/1777> ; DOI : <https://doi.org/10.4000/jtei.1777>

For this publication a Creative Commons Attribution 4.0 International license has been granted by the author(s) who retain full copyright.

What is TEI Conformance, and Why Should You Care?

Lou Burnard

ABSTRACT

The recommendations of the Text Encoding Initiative (TEI) seem to have become a defining feature of the methodological framework of the Digital Humanities, despite recurrent concerns that the system they define is at the same time both too rigorous for the manifold variability of humanistic text, and not precise enough to guarantee interoperability of resources defined using it. In this paper I question the utility of standardization in a scholarly context, proposing however that documentation of formal encoding practice is an essential part of scholarship. After discussing the range of information such documentation entails, I explore the notion of conformance proposed by the TEI Guidelines, suggesting that this must operate at both a technical syntactic level, and a less easily verifiable semantic level. One of the more noticeable features of the Guidelines is their

desire to have (as the French say) both the butter and the money for the butter; I will suggest that this polymorphous multiplicity is an essential component of the system, and has been a key factor in determining the TEI's continued relevance.

INDEX

Keywords: conformance, standardization, interoperability,

1. What Are Encoding Standards Actually For?

- 1 As the old joke says, the good thing about standards is that there are so many to choose from. You can choose to follow a dictatorial, centrally imposed, we-know-what's-best-for-you (WKWBFY) encoding method, like Microsoft Word. You can choose to follow a handcrafted, idiosyncratic, we-know-what-we're-doing (WKWWD) kind of encoding standard made up and maintained by the leading lights of a particular research community, like [EpiDoc](#).¹ Or you can just go ahead and do your own encoding thing, which I like to characterize as the nobody-understands-my-problems (NUMP) kind of standard. In academia, there's a good argument for each of these approaches; indeed it has been suggested that different components of the TEI itself accord differing priority to each of these. The TEI header, for example, can be used in a highly prescriptive WKWBFY manner, while users of the TEI proposals for feature-structure analysis must be assumed to Know What Is Best for Them. And every customizer of the TEI faced with a large amount of ambiguously encoded legacy data, or an intransigent user community, must be grateful that in some aspects it permits a NUMPy approach (the survival of both vanilla and "flavored" `<div>` elements such as `<div1>` and `<div2>` being a notable example). For each approach to standardization has its merits. WKWBFY saves a lot of time and effort reinventing the wheel and ensures that your work will be processable and usable in at least one application environment: the downside is that you may not want or like the world view that the system embodies, but you can't change it. WKWWD probably means you are dealing with congenial and familiar views and are guaranteed respect within your community, but no one outside that community will know what to do with your stuff, and you may be a bit limited if you want to push the boundaries of knowledge or praxis within it. And, of course, NUMP guarantees

you the luxury of making all your own decisions, getting everything just the way you want, but consequently not only risking isolation from your peers but also having to spend lots of time and effort doing technical work that has nothing to do with your real scholarly preoccupations.

- 2 When the choice is so hard to make, it may be a good idea to reconsider the motivation for making it in the first place. What do we actually gain from adopting an explicit encoding standard? What scholarly advantage is there in formally defining the formats of our digital re-presentations of cultural artifacts? We may do it simply in order to be seen to be ticking the right boxes in a funding agency's list of criteria; we may do it because our elders and betters have told us we should; we may do it because we know no better. Such considerations, though intellectually less persuasive, may play a more significant part in enlarging the community of standards-conforming users than motivations derived from a consideration of scholarly utility. But it still seems useful to ask the question: how does the use of explicit standards in the markup of digital resources contribute to the success or failure of a scholarly enterprise?
- 3 First, I suggest, we should not forget that the application of markup is an inherently scholarly action which, since the goal is to express a scholarly interpretation, is an inherently hermeneutic activity. The choice of markup vocabulary is not an arbitrary one, but one with consequences. Any given choice may make it harder to express a truth about a document or a document's intentions; it may make it easier to say something which is convenient, but false. To dismiss as "mere semantics" concerns about the proper application of markup is thus to embark upon a very dangerous path, if of course you share my belief that every scholarly encoding should truthfully represent without convenient distortion a scholarly reading.
- 4 Second, if the function of markup is to express an interpretation, then the markup language itself should as far as possible eschew ambiguity. Markup defines and determines the interface between human interpretation and algorithmic processing. It determines what an algorithm has at its disposal to work on yet it is frequently (if not necessarily) the product of a nonalgorithmic human interpretation. Life is complicated enough without introducing additional fuzziness and inconsistency into the processing stack. We would like to live in a world where two equally well-informed observers looking at the same encoding will reach similar or identical conclusions as to the interpretations which occasioned that encoding. We would also like to be confident that two equally well-informed encoders, considering the same textual phenomenon, and having the same

interpretation of it, will encode that interpretation in the same way. (This is not, of course, the same as wishing that all well-informed encoders should reach the same interpretative conclusions about a given text. Quite the contrary.) Consequently we desire the claims embodied by a marked-up document to be formally verifiable in some way, even though there may be limits to the extent to which that can be achieved. Verifiability implies the existence of some formal definition for the markup language, against which instances of it can be checked, preferably automatically. Talking solely of XML documents, we would prefer them to be not just “well-formed” but also “valid.” If two documents are both valid with respect to the same XML schema, we can at least have some confidence that they express comparable interpretations: we have reason to believe they will talk about the same things in the same terms.

- 5 The full validation of scholarly markup, however, requires more than simple XML validity since a marked-up document has intention beyond what an XML schema can express. A typical XML schema will allow me to say that the XML element `<p>` must appear within the XML element `<div>` and not the reverse, but it will not easily let me say that the content of my `<p>` elements corresponds with a paragraph of text rather than, say, a page or a potato. For that information, the user must consult project-specific documentation, which should spell out how exactly the intentions behind this set of encoded documents are realized by its surface components (such as the start and end tags and the attribute value pairs). In the absence of that documentation, all we have are expectations based on knowledge of habitual practice.
- 6 Third, therefore, we need to complement the automatic validation of our markup with semantic checks which, in our present state of knowledge, are not automatable, and require human judgement. It is no coincidence that SGML, the ancestor of XML, was produced by a lawyer: the rules embodied by an SGML DTD, like those in the statute book, must be interpreted to be used. In the field of the law, the statute book is completed by precedents; in the case of an XML schema used by a broad community such as the TEI, the rules incarnated in the TEI Guidelines are completed by the practice of those using them, whether we are thinking about the Guidelines as a whole, or the customizations of them used by individual projects. As discussed further below, a TEI customization expresses how a given project has chosen to interpret the general principles enumerated by the Guidelines, as well as formally specifying which particular components of the Guidelines it uses. It also provides ample opportunity, through documentation

and exemplification, to guide a human judgement as to the way in which the markup should be understood, and therefore the extent to which different datasets using it can be integrated or rendered interoperable.

2. How Are Encoding Standards to Be Documented?

- 7 As a minimum, the documentation of an encoding language has to be able to specify the same things as a schema does, including the names of the elements and attributes used, their possible contents, how elements may be validly combined, and what kinds of values are permitted for their attributes. Currently available schema languages do not provide an entirely identical range of facilities of this kind, nor do they conceptualize the validation of documents in exactly the same way, but they are in sufficiently broad agreement for it to be possible to model the information they require using a simple XML language, which now forms a part of the TEI tagset documentation system, referred to here and elsewhere as ODD: an abbreviation for “One Document Does it all” (see further [Rahtz and Burnard 2013](#); [Burnard and Rahtz 2004](#)).
- 8 Of course, if schema models were all that ODD supported, it would be hard to persuade anyone to use it. The full ODD language provides for much more than the basic information required to create a schema model. For example, a full TEI element specification may contain:
1. a canonical name for the element, together with explanatory glosses for the name in various languages; alternative names in other languages; equivalents in other markup schemes;
 2. at least one summary description of the meanings and usages intended for the element;
 3. information about the element and attribute classes to which the element belongs;
 4. information about the element’s content model;
 5. formal specifications for any constraints additional to those expressed by the content model;
 6. a list of specifications for any attributes defined as local to the element rather than being inherited from an attribute class;
 7. formal specifications for the recommended processing model applicable to the element;
 8. annotated examples of usage;
 9. additional commentary or usage notes;

- 10 a list of references to the chapter of the Guidelines text where the element is discussed more fully.
- 9 A TEI specification therefore potentially provides data which can be used to facilitate many different markup-related processes in a more productive way—making it possible, for example, to provide context-sensitive help in different languages, to generate formal schema specifications in different schema languages automatically, to provide user-oriented tutorial manuals, to develop more intelligent data entry and validation systems, or simply to act as a point of entry into the canonical TEI documentation.
- 10 A criticism sometimes made of XML schemas in general and the TEI in particular is that they encourage an excessively reductionist perspective. Robinson (2009) is not alone in asserting that “the concept of what ‘text’ is, upon which (for instance) the Text Encoding Initiative principles are based ... is positivist, overconfident, simplistic and neglects the materiality of actual text instances.” TEI markup is believed to convey what a text “really is” (De Rose et al. 1990), discarding as secondary distractions all the rugosities of real text such as choice of typeface or accidentals of printing. But when choosing and defining a markup scheme, different projects will legitimately disagree as to which aspects are distractions and which are essential. It may after all be that typographic accidentals are precisely the focus of scholarly attention, as for example when transcribing early print editions, to say nothing of manuscripts. Comparatively recent developments of the TEI have attempted to address this concern by enriching the available element set, enabling it to represent (for example) digital facsimiles, detailed manuscript transcriptions and descriptions, or genetic editions. The cost of this enrichment is evidently that the TEI model becomes more complex, so much so that it is doubtful whether it makes any sense to talk of a single TEI model, even though some parts of it seem to be universal. The benefit of this enrichment is, equally evidently, the ability it confers for a far broader community of users to define models precisely fitted to their particular textual perspective, even if those models differ widely in the importance they attribute to particular textual features. Those for whom the digital edition is a carefully constructed textual reading or set of readings can use the TEI to encode them, just as

effectively as those for whom it is a collection of scrupulously represented and analyzed page images. Both communities benefit, moreover, from the availability of a set of common concepts which do not need to be redefined or disputed.

- 11 It is often claimed that by focusing on the platonic essence of some collection of data, a data model will equally effectively facilitate many possible applications for those data. But in the real world, a compromise must be found: in the absence of any engagement at all with the way the data being modeled is intended to be used, a model risks being of theoretical interest only. The “processing model” recently added to the TEI ODD language is an example of a facility provided by that system which may be said to help redress that balance by formally specifying the kind of processing that the encoder considers appropriate for a given element. The processing model for an element might, for example, just indicate the class of formatting appropriate for it; or it might indicate that this is one of the so-called “janus” elements which present alternative encodings for the same phenomenon. In either case, the intention is to simplify the task of the application developer faced with a specific customization of the TEI, by enlarging the scope of available information beyond what is provided by an XML schema. Such documentation will also clearly benefit the person attempting to curate the data for the long term. One of the many partially fulfilled promises of the SGML revolution was that by abstracting data description away from data processing, data longevity would be assured. The experience of those who have tried to reuse existing TEI resources suggests that this is true, if only up to a point. One of the take-home messages from ambitious reuse projects such as Project Monk (Unsworth and Mueller 2009) appears to be that ease of reuse is directly proportionate to the availability of precise documentation concerning the way a project used or intended to use the TEI.
- 12 We conclude, therefore, that the way a data model is recorded and documented is likely to be of critical importance in determining its long-term usefulness as well as its immediate effectiveness. Today’s digital projects are complex and sometimes overengineered constructs; maintaining a clear and accurate record of the data models on which they are based is correspondingly necessary.

3. The Importance of Customization

- 13 As suggested above, the richness of today's TEI makes it almost essential to customize it for any given application, by selecting from the available specifications. To facilitate that task, the specifications are grouped together, both physically into named "modules," and logically into named "classes." Each module contains a number of related declarations, and modules can be combined as necessary, though in practice there are a few modules which provide components needed by almost any encoding. In earlier versions of the Guidelines, a distinction was made between modules which provided components specific to a particular kind of document (the "base" tagsets) and those which provided components specific to a particular kind of analysis (the "topping" tagsets). The idea was that a schema would typically use a single base and multiple toppings, though it was also possible to combine multiple bases. This, the so-called pizza model, did not survive into TEI P5, where all modules are considered equal. A class, by contrast, is an abstract object to which elements point in order to express their semantic or structural status. Two kinds of classes are distinguished: model classes and attribute classes. The members of a model class share structural properties, in particular the locations in a content model where they are permitted; the members of an attribute class share identically defined attributes. A customization may define new classes, delete existing ones, or modify the membership of a class, though these are facilities that can have unexpected consequences, as we discuss below. In SGML-based versions of the Guidelines, classes were represented by parameter entities which could be modified by a document instance, thus providing extension points for the encoding scheme. In RELAX NG, classes are represented as patterns, with similar capabilities.
- 14 A customization which just specifies a selection of modules will overgenerate, not only in the sense that the resulting schema will contain specifications for components that will never be used, but also because the TEI often provides multiple ways of encoding the same phenomenon. For example, the TEI provides three elements for the representation of a traditional bibliographic record: `<bibl>`, `<biblStruct>`, and `<biblFull>`; these elements differ in their internal syntax, but are semantically identical. Similarly, the core module provides a handful of elements (such as `<foreign>`, `<emph>`, and `<soCalled>`) for signaling the function associated with visual distinctions such as italicization or quotation marks, while also providing a way of simply signaling the fact of visual salience or highlighting itself by means of the element `<hi>`. Any or all of these may

use any or all of three attributes, `@rend`, `@rendition`, or `@style`, each of which offers a quite different way of representing how exactly the source is visually or otherwise salient. Similarly, the TEI `att.dataable` attribute class provides two distinct sets of attributes for normalizing dates and times, one conforming to W3C, the other conforming to ISO.² Plus, for good measure, a third subclass called `att.dataable.custom` allows the encoder to use their own conventions. The TEI is scrupulously agnostic even about how a TEI document itself is to be constructed: the classic TEI document comprises a TEI header and a transcribed text; the transcribed text may, however, be combined with a set of digitized images or replaced by one. It is also possible to replace (or complement) the traditional text transcription (which aims to capture the logical organization of the source document) with a “source-oriented” transcription which captures just its physical organization and eschews other interpretive gestures. And there are plans to add a further parallel component to contain annotations made upon the text in a “stand-off” manner (see further [Pose et al. 2014](#)).

- 15 This multiplicity of choice can be bewildering and may seem absurd. Yet every element and attribute in the TEI Guidelines is there because some member of the scholarly community has plausibly argued that it is essential to their needs; where there is a choice, therefore, it is not because the TEI is indecisive, it is because all of the available options have been considered necessary by someone, even if no one (except perhaps those blessed with the task of maintaining the TEI) ever considers all of them together.
- 16 A project wishing to use the TEI, and to document that usage accurately, is therefore obliged to proceed with caution. Just selecting a few promising modules is not necessarily the best approach: a selection must also be made from the components provided by those modules, since selecting everything available is a recipe for confusion. Those unwilling or inadequately resourced to make this effort can use one of the generic TEI customizations made available by the TEI itself ([TEI simplePrint](#),³ for example), or by specific research communities ([EpiDoc](#)⁴ is an excellent example; see also [Bodard 2010](#)). But adopting an off-the-peg encoding system is always going to be less satisfactory than customizing one that fits more precisely the actual needs of a project and the actual data modeled within it. (It is assumed that a data analysis of some kind is a necessary precursor to any digital project.)

- 17 Furthermore, it is painfully true that nothing in digital form is ever really finished. Almost inevitably, as a project evolves, things that should have been done differently will emerge. In the light of experience, it becomes much easier to change the list of available elements to match actual encoding practices more closely. Beginners often think it is better to allow almost any kind of content in their schema: an extreme case of this misapprehension leads people to use `tei_all` (the exhaustive TEI exemplar schema which includes the entirety of TEI features) for everything. It may well be that a project starts off a little uncertain about the kind of data it will have to be able to handle. But as an encoding project matures, these uncertainties disappear and project-specific praxis becomes better understood. The cost-benefit ratio of allowing for the unforeseen begins to change. Every element in your schema is another element to explain to the encoders, another element to document and find examples for, and another element whose usage needs to be checked for consistency. It is also another element that the poor overworked software developer writing code to process the encoding has to be prepared to handle.
- 18 Similar considerations apply to attributes, and in particular to their ranges of values. At the outset of a project, it may have been impossible to predict what values would be appropriate for some attribute, and hence the initial model will have allowed anything. The price of this laissez-faire policy is that in the absence of guidance, encoders will supply widely varying values, for example "centre", "centered", or "middle", all meaning (probably) the same thing. Once it becomes clear which values are appropriate, it is better to provide such guidance, by adding a `<valList>` to the ODD, even if this entails some additional work cleaning up existing data.
- 19 Customization is very often a simple matter of selection, or formally speaking a subsetting operation. For example, a customization which specifies that attribute values be taken from a closed list of possible values, rather than being any token of the appropriate datatype, is a subsetting operation: the set of documents it considers valid is a pure subset of the set of documents considered
- 20 The diagram in [figure 1](#) represents the variety of possible customizations.

Figure 1. Varieties of Customization.

- 21 Each of the shapes here may be understood to represent three different things:
- an ODD: that is, a collection of TEI specifications constituting a customization
 - a formal schema generated from that ODD, and its natural language documentation
 - the set of documents considered valid by that schema
- 22 The TEI provides a monolithic unmodified schema called `tei_all` which contains all of the elements, classes, macros, and other components defined by the TEI. As noted above, for all practical purposes a user of the TEI must make a selection from this cornucopia, which I will call a “TEI customization,” represented as a named set of modifications and encoded by a TEI ODD. Of course there are many, many possible TEI customizations, each involving different choices of elements or attributes or classes, but there are at least two different kinds of customization: a *TEI subset* and a *TEI extension*. (In proposing this terminology, I am reinventing a distinction proposed by David Birnbaum [Birnbaum 2000, esp. sec. 5.1] which talks of modifications as “supersets” or “subsets.”)

- 23 When a set of modifications results in a schema which regards as valid a subset of the documents considered valid by `tei_all`, I will call this a “TEI subset”; the Guidelines also identify this as a *clean modification*. Where this is not the case, I propose the term “TEI extension,” which the Guidelines identify as an *unclean modification*. A customization which adds new elements or attributes, or one in which elements are systematically renamed, cannot result in a subset, because the set of documents that the schema generated from it will consider valid is not a proper subset of the documents regarded as valid by the `tei_all` schema.
- 24 A change to the content model of an existing TEI element may or may not result in a TEI subset. For example, if `tei_all` does not specify an order for the child elements of some content model, a customization which does constrain that order will be a TEI subset: every document it considers to be valid is also valid according to `tei_all`. The reverse is not the case, however: if `tei_all` does specify an order, a customization which relaxes that constraint will result in a schema that considers valid some documents considered invalid by `tei_all`; it is therefore a TEI extension. Removing an element from an attribute class will result in that element losing the attributes supplied by the class which (on the assumption that none of them is mandatory) will result in a subset. Removing an element from a model class will similarly (in most cases) mean that the element ceases to be available in the content of other elements, and hence also usually results in a subset. Adding an element to a model class in which it did not previously figure, however, will usually result in a TEI extension. (Further examples of typical kinds of modification are given in the Guidelines: [TEI Consortium 2018, chap. 23](#).)
- 25 TEI extensions which include TEI elements or attributes whose properties or semantics have been significantly changed are expected to place those elements or attributes in a different namespace. Such elements should, as far as possible, be included in existing content models by making them members of existing TEI classes, rather than by explicitly modifying the content model of an existing element. On the face of it, this means that any element referencing a new element will have a different content model, and should therefore be in a different namespace too. And the same ought to apply to its parent elements, and so on up to the TEI element itself. Fortunately, there is a nuance of detail which means we do not need to invoke this “turtles all the way up” scenario, provided content models are defined not in terms of specific elements but with reference to model classes. A class reference will be dereferenced to a specific set of elements only when an ODD is

converted to a schema; this is necessary because the set in question will depend on which elements are available in the customization. Any element, including one from a non-TEI namespace, may claim membership in a TEI model class and hence legitimately appear in the content of a TEI element referencing that class.

- 26 I argue in the next section that TEI conformance is not simply a matter of validity against a schema. Nevertheless, there are a few hardwired rules built into the TEI model, which the customizer ignores at their (or rather, their potential audience's) peril.
- 27 For example, a TEI Header really *must* have a title statement containing at least one title, along with a publication statement and source description, even if the latter two have no significant content. A TEI `<text>` element really *must* contain a `<body>` element. TEI `<div>` elements really *must* nest correctly within one other. The structural classes in terms of which content models are defined really *must* be respected: hence one `<p>` cannot contain another, and a phrase-level element such as `<hi>` cannot contain a block-like element such as `<p>`.
- 28 Some of these restrictions are the subject of regular debate on TEI-L and elsewhere, but for the most part they are, in my view, integral parts of the TEI model. It is a part of the definition of a TEI `<div>` that once you have encountered another nested `<div>` within it, only `<div>` elements at the same hierarchic level are permitted until it finishes; a non-tessellating division element might well be useful, but if one is defined it must be distinguished clearly from the existing TEI `<div>`, for example by placing it in a different namespace (assuming that this particular aspect of the TEI abstract model remains unchanged).
- 29 Breaking these rules may have unexpected consequences. For example, a customization which removes the element `<title>` completely will result in a schema in which no TEI `<teiHeader>` element can ever be considered valid, since the mandatory components of the TEI Header are an essential part of it; a TEI Header which lacks them is a different kind of object, and should not present itself as being something which it is not.

4. What Is TEI Conformance?

- 30 Umberto Eco remarks that “a novel is a machine for generating interpretations” (Eco 1984, 2). We might say that the TEI is a machine for generating schemas to formally represent such interpretations. However, just as not all interpretations of a novel have equivalent expository

force, so not all TEI customizations are of equal effectiveness or appropriateness; indeed, what is effective or appropriate for one purpose may not be for another. A customization explains and defines a view of what it is meaningful and appropriate for a given project to assert about a set of instance documents. It does this by reference to the very large pre-existing range of concepts distinguished by the TEI, selecting from that range the particular distinctions it wishes to make, possibly modifying some of them, possibly adding to them. I suggest that our assessment of the “appropriateness,” or conformance, of a given customization should take into account the way in which that customization is expressed and the claims it consequently makes about its understanding of concepts originally enumerated by the TEI.

- 31 There are also good pragmatic grounds for wanting to know how a given customization has modified the TEI definitions. Such knowledge enables us to make comparisons among different customizations, to assess their relative closeness to the original Guidelines, and to determine what might be necessary to make documents using those different customizations interchangeable, if not interoperable. As Martin Holmes (2016) and others have pointed out, the pursuit of unmediated interoperability among TEI documents is largely chimerical, whereas the information provided by the documentation of a TEI customization will often be all that is needed to make documents using that customization interchangeable.
- 32 Beyond pragmatic considerations, however, a definition of TEI conformance, and some rules of thumb for assessing it, are surely needed if the TEI Guidelines are to be more than increasingly stale dogma. I suggest that the most useful definition of TEI conformance is one that takes into account the original design goals and recommendations of the Initiative (Sperberg-McQueen and Burnard 1988), which are explicitly formulated to benefit a very broad range of applications and disciplines, rather than to impose a single one-size-fits-all standard. It follows that an assessment of TEI conformance involves more than simply checking whether the customization is a subset of `tei_all` or an extension of it in which all non-TEI components have been clearly identified as such. It must also take into account the extent to which the encoding respects the TEI semantic model. Where the encoding uses TEI-defined elements, these should represent the concepts associated with those elements by the TEI Guidelines. Where an encoding enlarges that set of concepts, it should not misuse TEI elements to do so, but make explicit that it addresses concepts not addressed by the TEI, for example by placing elements in a non-TEI namespace.

- 33 In assessing conformance, there is a natural tendency to attach particular importance to validity against a schema, since this is something which can be automatically tested, whereas checking the semantic validity of an encoding is not in principle automatable. Validation of a document which uses a TEI extension may additionally require the use of a namespace-aware validator such as `ONVDL`,⁵ which can validate different parts of an XML document against possibly many different schemas, using the Namespace-based Validation Dispatching Language, defined as part 4 of the ISO standard for Document Schema Definition Languages (ISO 2006). This is one reason why validity against `tei_all` has limited significance in assessing the conformance of a customization, other than to determine whether it is a TEI subset or a TEI extension. The TEI was designed to facilitate customization both by subsetting and by extension; either process therefore has the potential to result in something which should be considered “conformant.”
- 34 Earlier versions of the P5 Guidelines introduced a notion of “TEI conformable” or “algorithmic conformance” to identify TEI extensions which might be placed at the “acceptable” end of a notional scale of conformance. “A document is said to be TEI Conformable if it is a well-formed XML document which can be transformed algorithmically and automatically into a TEI Conformant document as defined above without loss of information” (TEI Consortium 2008, sec. 23.3). A customization which systematically provides alternative identifiers for all the TEI elements it uses by means of the `<altIdent>` element provided for this purpose is clearly more “conformable” than one which simply redefines a few elements with different names, but with the same semantics and the same content models as the existing ones. A customization which defines a new element (say `<botanicalName>`) and adds it to an existing class (`model.nameLike`, for example) will be more “conformable” if it also provides an explicit TEI mapping for it, using the `<equiv>` element in its specification to indicate that this element is equivalent to `<name type="botanical">`. However, despite its initial attractiveness, the notion of “conformability” is impossible to define exhaustively and precisely, since anything can be algorithmically converted to something else, given sufficient ingenuity, and it no longer features in the Guidelines discussion of conformance.
- 35 Nevertheless, if the only distinction we can make as regards conformance is between a TEI subset (which is probably ipso facto conformant, though not necessarily) and a TEI extension (which is probably ipso facto nonconformant, though not necessarily), the notion is not a very helpful one. Intuitively, it seems evident that some modifications are closer in spirit to the TEI conceptual model

than others, even if they result in a schema which is not a TEI subset as previously defined. At the “more acceptable” end of the scale we might place the case of systematic renaming already cited; another might be the addition of additional attributes to an existing element, where that does not result in a conflict of any kind. For example, not all elements are members of the class `att.typed`, and thus many cannot be subcategorized. A customization in which (say) the `<address>` element gains a `@type` attribute so that the encoder can distinguish postal addresses from email addresses seems entirely innocuous. A processor which knows about the TEI element `<address>` will not go badly astray if it simply ignores the new attribute. If, however, the new attribute has a value such as “machine” or “IP”, we might begin to feel that the concept of “address” has shifted somewhat, and an uninformed TEI processor might well be at a loss to deal with it. Subcategorization of this kind should not be a back door to redefining the meaning of an element completely.

- 36 A more problematic case of nonconformance would be where a generic subcategorization attribute such as `@type` is used to make a distinction already made by an existing more precise attribute. For example, `<title>` has both `@level` and `@type` attributes; using the latter to specify the former would be a nonconformant usage, even though, in this case, no extension is involved. It is analogous to inventing a new non-TEI element that duplicates the function of an existing TEI element. Because the TEI is hospitable to other XML vocabularies, an extension using elements from another scheme may easily introduce such duplications. For example, if Dublin Core elements were embedded directly in the TEI Header, they would duplicate the function of many existing TEI elements, with consequent integrity problems, even though the DC elements might be clearly labelled as non-TEI, or even embedded within the TEI `<xenodata>` element.
- 37 The ability to extend the range of encodings supported by the TEI simply and straightforwardly remains a fundamental requirement for a scheme which is intended to serve the needs of research. This requirement has several important benefits:
- it enables the TEI to integrate with comparative ease other specialized XML vocabularies, such as MathML, SVG, or most recently MML;
 - it facilitates and encourages the development of new TEI components by the broader community;
 - it simplifies the task of interchange by reducing the possibility of ambiguous or incoherent encoding.

- 38 This polytheoricity underlies the TEI's apparent complexity, and is also a major motivation for the requirement that a modification should use namespaces in a coherent manner: in particular, that elements not defined by the TEI, or TEI elements whose definition has been modified to such an extent that they arguably no longer represent the same concept, should not be defined within the TEI namespace. Of course, reasonable people may reasonably disagree about whether two concepts are semantically different, just as they may disagree about how to define either concept in the first place. That is part of what Darrell Raymond (in an unpublished draft of [Raymond, Tompa, and Wood 1996](#)) memorably called the "hellfire of ontology" into which the descriptive markup project has plunged an entire generation.
- 39 As a simple example of these ontological anxieties, consider the TEI `<stamp>` element. This is intended, if one is to judge from the examples in the Guidelines, to document marks impressed on a manuscript or incunable with the aid of a rubber stamp or similar object, typically to indicate ownership. But what of marks impressed on a letter to indicate the time and date it was posted (the "postmark")? What, indeed, of the little paper sticker affixed to a postcard or envelope to indicate that postage has been paid, the postage stamp? Is either of these semantically close enough to the TEI's existing `<stamp>` for us to use that element to document them? If a postage stamp is not a kind of TEI `<stamp>`, maybe it is a kind of TEI `<seal>`? A TEI customization designed for a collection of transcribed postcards might choose to make use of either of these existing elements for the purpose; or it might choose to define its own new element. A TEI application searching blindly through an archive for information about seals or stamps (according to the TEI definitions) will be misled by either of the first modifications, whereas the presence of a new element should not confuse it, as previously noted. This suggests, perhaps paradoxically, that the use of a non-TEI element results in a more conformant document than the repurposing of existing, semantically related, elements.
- 40 As a further example, consider the Dublin Core `<dc:title>` element. This has the same semantics and serves the same function as the TEI `<title>` which is a mandatory component of the TEI `<titleStmt>` in the TEI Header. A modification which redefines the TEI `<titleStmt>` to require a `<dc:title>` instead of a `<title>` might plausibly be argued to respect the TEI conceptual model, even though it is clearly invalid with respect to a TEI schema.

- 41 The ability to use explicitly labeled non-TEI elements in a TEI modification may be seen as an important coping mechanism, enabling the TEI community to experiment with changes to the usual ways of doing things and to enlarge the scope of the scheme in a controlled and nondisruptive way. I have argued elsewhere (Burnard 2013) that one reason for the TEI's continued longevity is precisely its ability to mutate and evolve. This ability is not without a price: the creation of a customization does require some knowledge of the whole architecture, and some technical expertise. For those already expert in database or non-TEI XML technologies and for the novice alike, the effort to maintain TEI conformance may seem an unnecessary additional hurdle.
- 42 These difficulties do not, however, invalidate the general principle that TEI conformance should entail a respect for the consensus, just as much as it facilitates autonomy. As we have shown, even in the case of a customization which has eschewed extension and appears to be a straightforward TEI subset, an assessment of TEI conformance involves attention to some constraints which are not formally verifiable. I conclude by suggesting that conformance requires attention to two important if largely unenforceable requirements of "honesty" and "explicitness."
- 43 By "honesty" I mean that elements in the TEI namespace must respect the semantics which the TEI Guidelines supply as a part of their definition. For example, the TEI defines an element <l> as containing "a single, possibly incomplete, line of verse."⁶ If an encoding distinguishes verse and prose, it would be dishonest to use this element to mark line breaks in prose, since to do so would imply that the element contains verse rather than prose. Most TEI elements are provided in order to make an assertion about the semantics of a piece of text: that it contains a personal name rather than a place name, for example, or a date rather than a number. Misapplying such elements is clearly counterproductive. (Honestly made misreadings are of course entirely forgivable: an encoding always asserts an interpretation, not the absolute truth of that interpretation.)
- 44 By "explicitness" I mean that modifications should be properly documented, preferably by means of an ODD specifying exactly how the TEI declarations on which they are based have been derived. (An ODD need not of course be based on the TEI at all, but in that case the question of TEI conformance does not arise.) The ODD language is rich in documentary components, not all of which are automatically processable, if only because their processing is not fully specified (the <equiv> and <altIdentifier> elements, for example). But it is usually much easier to determine

how the markup of a set of documents should be interpreted or processed from an ODD than it is from the many pages of human-readable documentation needed to explain everything about an idiosyncratic encoding scheme.

45 To summarize and conclude, I suggest that we should say of a document that it is “TEI conformant” if and only if :

- it is a well formed XML document; and
- it is valid against one or more schemas, which may be either a TEI subset or a TEI extension; and
- its usage of elements in the TEI namespace is compatible with the intended function of those elements as defined by the TEI Guidelines; and
- its usage of the TEI markup scheme is fully described by a TEI-conformant ODD or analogous documentation.

46 The purpose of these rules is to make interchange of documents easier. They do not guarantee it, and they certainly do not provide any guarantee of interoperability. But they make much simpler, for example, the kind of scenario envisaged by Holmes (2016) in which a richly encoded, highly personalized TEI encoding can be simply down-translated to other, possibly less expressive, semistandardized encodings for purposes of interchange. As more and more independent agencies undertake mass digitization and encoding projects, the risk of a new confusion of tongues—the threatened Tower of Babel which the TEI was specifically created to resist—has not retreated. A definition of conformance which relies on an enforced lowest common denominator standard (Dublin Core springs to mind) makes it hard to benefit from truly sophisticated and scholarly standards. One which promotes permissiveness and extensibility, as the TEI does, has to balance the sophistication of what it makes feasible with a clear and accessible definition of its markup. Unlike many other standards, the goal of the TEI “standard” is not to enforce consistency of encoding, but to provide a means by which encoding choices and policies may be more readily understood, and hence more easily made algorithmically comparable.

BIBLIOGRAPHY

- Birnbaum, David J. 2000. "The Relationship between General and Specific DTDs: Criticizing TEI Critical Editions." *Markup Languages: Theory and Practice* 3 (1): 17–53. Author's version, last revised July 5, 2000, available at <http://www.obdurodon.org/djb/tei-crit/>.
- Bodard, Gabriel. 2010. "EpiDoc: Epigraphic Documents in XML for Publication and Interchange." In *Latin on Stone: Epigraphic Research and Electronic Archives*, edited by Francisca Feraudi-Gruénais, 101–18. Lanham, MD: Lexington Books.
- Burnard, Lou. 2013. "The Evolution of the Text Encoding Initiative: From Research Project to Research Infrastructure." *Journal of the Text Encoding Initiative* 5. <https://journals.openedition.org/jtei/811>; doi:10.4000/jtei.811.
- Burnard, Lou, and Sebastian Rahtz. 2004. "RelaxNG with Son of ODD." In *Proceedings of Extreme Markup Languages 2004*. <http://conferences.idealliance.org/extreme/html/2004/Burnard01/EML2004Burnard01.html>.
- De Rose, Steven J., David G. Durand, Elli Mylonas, and Allen H. Renear. 1990. "What Is Text, Really?" *Journal of Computing in Higher Education* 1 (2): 3–26. doi:10.1007/BF02941632.
- Eco, Umberto. 1984. *Postscript to "The Name of the Rose"*. Translated by William Weaver. New York: Harcourt Brace Jovanovich.
- Holmes, Martin. 2016. "Whatever Happened to Interchange?" *Digital Scholarship in the Humanities* 32, Issue suppl_1 (1 April 2017): i63–i68. doi:10.1093/lc/fqw048.
- ISO (International Organization for Standardization). 2006. *Information technology — Document Schema Definition Languages (DSDL) — Part 4: Namespace-based Validation Dispatching Language (NVDL)*. ISO/IEC 19757–4:2006. Geneva: ISO. <https://www.iso.org/standard/38615.html>.
- Pose, Javier, Patrice Lopez, and Laurent Romary. 2014. "A Generic Formalism for Encoding Stand-off Annotations in TEI." <https://hal.inria.fr/hal-01061548/>.
- Rahtz, Sebastian, and Lou Burnard. 2013. "Reviewing the TEI ODD System." In *Proceedings of the 2013 ACM Symposium on Document Engineering (DocEng '13)*, 193–96. New York: ACM. <https://ora.ox.ac.uk/objects/pubs:434097>; doi:10.1145/2494266.2494321.

- Raymond, Darrell, Frank Tompa, and Derick Wood. 1996. "From Data Representation to Data Model: Meta-semantic Issues in the Evolution of SGML." *Computer Standards & Interfaces* 18 (1): 25–36. doi:10.1016/0920-5489(96)00033-5.
- Robinson, Peter. 2009. "What Text Really Is Not, and Why Editors Have to Learn to Swim." *Literary and Linguistic Computing* 24 (1): 41–52. doi:10.1093/lc/fqn030.
- Sperberg-McQueen, C. M., and Lou Burnard. 1988. "Design Principles for Text Encoding Guidelines. TEI ED P1." December 14, 1988, revised January 9, 1990. <http://www.tei-c.org/Vault/ED/edp01.htm>.
- TEI Consortium. 2008. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, edited by Lou Burnard and Syd Bauman. Version 1.1.0. Oxford: TEI Consortium. PDF and HTML versions available at <http://www.tei-c.org/Vault/P5/1.1.0/doc/tei-p5-doc/en/html/>.
- . 2018. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Version 3.4.0. Last updated July 23. N.p.: TEI Consortium. <http://www.tei-c.org/Vault/P5/3.4.0/doc/tei-p5-doc/en/html/>.
- Unsworth, John, and Martin Mueller. 2009. "The MONK Project Final Report." September 2. <http://www.monkproject.org/MONKProjectFinalReport.pdf>.

NOTES

- 1 Tom Elliott, Gabriel Bodard, Hugh Cayless, et al., *EpiDoc: Epigraphic Documents in TEI XML*, 2006–2017, accessed October 25, 2018, <http://epidoc.sf.net>.
- 2 The W3C data formats are as specified by *XML Schema Part 2: Datatypes Second Edition*; the ISO formats are specified by ISO/IEC 8601.
- 3 See Lou Burnard, Martin Mueller, Sebastian Rahtz, James Cummings, and Magdalena Turska, *An Introduction to TEI simplePrint*, n.p.: TEI Consortium, January 2017, accessed October 26, 2018, http://www.tei-c.org/Vault/P5/3.4.0/doc/tei-p5-exemplars/html/tei_simplePrint.doc.html.
- 4 Tom Elliott, Gabriel Bodard, Elli Mylonas, Simona Stoyanova, Charlotte Tupman, Scott Vanderbilt, et al., *EpiDoc Guidelines: Ancient documents in TEI XML* (version 9), 2007–2017, accessed October 26, 2018, <http://www.stoa.org/epidoc/gl/latest/>.
- 5 "oNVDL - Oxygen XML NVDL Implementation Based on Jing," Craiova, Romania: Syncro Soft, accessed October 26, 2018, <https://www.oxygenxml.com/onvdl.html>.
- 6 TEI Consortium. 2018. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Version 3.4.0. Last updated July 23. N.p.: TEI Consortium. <http://www.tei-c.org/Vault/P5/3.4.0/doc/tei-p5-doc/en/html/ref-l.html>.

AUTHOR

LOU BURNARD

Lou Burnard has been closely associated with the TEI since its inception, initially as European editor, as an elected member of the TEI Board and for many years as an active member of the TEI Council. Since retirement from Oxford University Computing Services, where he was responsible for the development of the Oxford Text Archive, the British National Corpus, and several other key projects in the Digital Humanities, he has continued to provide consultancy and training services to many research projects, notably in France.