What is TEI conformance, and why should you care?

Lou Burnard *2017-10-29*

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 talk 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 discussion of 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.

1 What are encoding standards actually for?

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 encoding method like using Microsoft Word. You can choose to follow a hand-crafted, idiosyncratic, we-know-what-we're-doing kind of encoding standard made up and maintained by the leading lights of a particular research community, like Epidoc. Or you can just go ahead and do your own encoding thing, which I like to characterize as the nobody-understands-my-problems kind of standard. In academia, there's a good argument for each of these flavours. 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 kind of 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 tecchie things that have nothing to do with your real scholarly preoccupations.

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 artefacts? 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. But none of these can be considered well-founded motivations. How does the use of explicit standards in the markup of digital resources contribute to the success or failure of a scholarly enterprise using them ?

Firstly, I suggest, we should not forget that the application of markup is an inherently scholarly act: it expresses a scholarly interpretation. It is a hermeneutic activity. Our choice of markup vocabulary is therefore not an arbitrary one. It has consequences. It 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 that is you share my belief that every scholarly encoding should truthfully represent without convenient distortion a scholarly reading.

Secondly, 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 algorithmic processing and 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, as far as possible, we expect the claims embodied by a marked up document to be formally verifiable in some way. Verifiability implies the existence of some formal definition for the markup language, against which productions using it can be checked, preferably automatically. Talking solely of XML documents, we would prefer them to be not just 'well-formed' but also 'valid'.

Scholarly markup however requires more than simple XML validity. 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 and not the reverse, but it won't easily let me say that the content of my <p> elements should correspond to a paragraph of text rather than, say, a page or a potato. For that information, I will need to consult the project-specific documentation, which should spell out how exactly the intentions behind this set of encoded documents.

Thirdly, therefore, we need to complement the automatic validation of our markup with semantic controls which, in our present state of knowledge, are not automatable, and require human judgment. 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 must be completed by practice of those using them, whether we are thinking about the Guidelines as a whole, or the customizations of them used by individual projects. A TEI customization expresses how a given project has interpreted 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 judgment 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, a point to which we will return.

2 How are encoding standards to be documented?

As a minimum, the documentation of an encoding language has to be able to specify the same things as a schema does: the names of the elements and attributes used, their possible contents, how elements may be validly combined, what kinds of values are permitted for their attributes, and so on. The schema languages currently available to us do not provide an entirely identical range of facilities of this kind, nor do they conceptualise 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 ODD, the TEI tagset documentation system. Of course, if schema models were all that ODD supported, it would be hard to persuade anyone to use it. The full ODD language of course provides for much more than the basic information required to create a schema model, as I think it safe to assume that my present audience is well aware.

A criticism sometimes made of XML schemas in general and the TEI in particular is that their focus on data independence leads to a focus on the platonic essence of the data model at the expense of an engagement with the rugosities needful when making the data actually useful or usable. The 'processing model' is another recent addition to the TEI ODD language intended to redress that imbalance by formally specifying the kind of processing that the encoder considers appropriate for a given element.

A TEI customization is made 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 one or two modules providing components which are needed in almost any encoding. A class by contrast is an abstract object to which elements point in order to express their semantic or structural status.

A customization which just specifies a bunch of modules will over-generate, 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. The TEI core module provides both <bibl> and <biblStruct> as ways of representing a bibliographic record; the same module provides a handful of elements for signalling the function associated with visual distinctions such as italicisation or quote marks, while also providing a way of simply signalling the fact of visual salience or highlighting itself. Any or all of these

may use any or all of three quite different ways of representing the form of that visual salience in the source, provided by the attribute class att.global.rendition. Similarly, the TEI att.datable attribute class provides two distinct sets of attributes for normalising dates and times, one conforming to W3C, the other conforming to ISO. Plus, for good measure, a third sub-class called att.datable.custom which allows the user to specify 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 eschewing other interpretive gestures. And there are plans to add a further text-level component to contain annotations made upon the text in a 'standoff' manner.

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.

A project wishing to use the TEI is therefore obliged to consider carefully how to use it. Just selecting a few promising modules is not necessarily the best approach: you will also need to select 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 or other of the generic TEI customizations made available by the TEI itself (TEI Simple Print, for example), or by specific research communities (Epidoc is an excellent example). But it is my contention that 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 your project and the actual data you have modelled within it. (You did do a data analysis before you started, didn't you?).

And whether or not you did, it's painfully true that nothing in digital form is ever really finished. It's almost inevitable that as your project evolves, you will come across things you would do differently if you could start all over again. In the light of experience, you may well want to change the list of available elements to match more closely your actual encoding practices. Beginners often think that it's better to allow almost any kind of content in their schema: an extreme case of this misapprehension leads people to use TEI_all for everything. It may well be that your project started out a bit uncertain about the kind of data it would 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 you allow for in your schema is another element you need to explain to your encoders, another element you need to document and find examples for, and another element whose usage you need to check for consistency. It's also another element that the poor over-worked software developer has to be prepared to handle.

Similar considerations apply to attributes, and in particular to their range of values. At the outset you may not have been sure what values to permit for the *@foo* attribute on your `<bar>` elements, so you allowed anything. Now you have discovered that some of your encoders gave this attribute the value 'centre', others used 'centered', and yet others used 'middle', all meaning (probably) the same thing. Now that you know which values you want, you will want to add a `<valList>` to your customization to enforce them, even if this entails some additional work cleaning up existing data.

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 considered valid by that customization is a

pure subset of the set of documents considered valid by a schema lacking that particular customization. But this may or may not be true of a modification which changes the datatype of an attribute : for example a change from string to date is a subsetting operation, whereas the reverse modification, (from date to string) is not.

And it is easy to think of apparently benign and useful modifications which inevitably result in an extension, rather than a subset. For example, a modification may provide an alternative identifier for an existing element or attribute, for example to translate its canonical English name into another language. A modification may change the class memberships of an existing element, so that it acquires attributes not previously available, or so that it may appear in contexts where it previously could not. A modification may change the content model of an element to permit different child elements, or so that existing children elements may appear in a different order or with different cardinalities. And of course a modification can readily define entirely new elements, macros, classes or attributes, and reference them from existing TEI components, within certain limits. The following diagram is intended to demonstrate some of these notions.

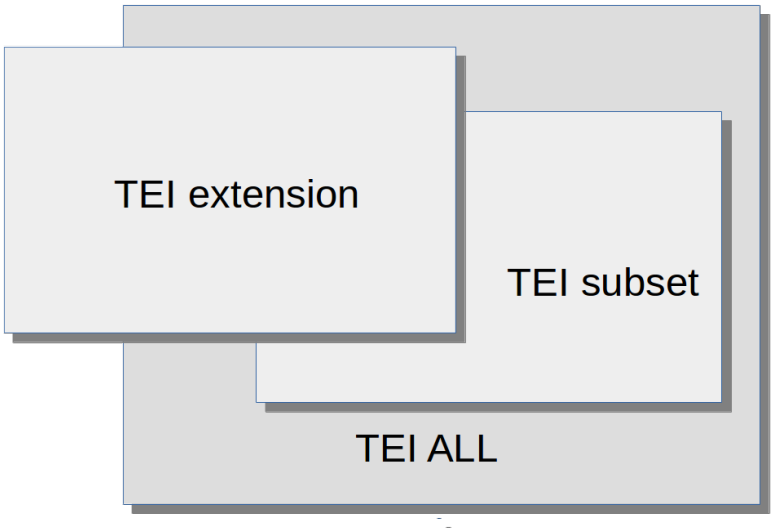

Figure 1: Varieties of Customization

Each of the shapes here may be understood to represent three different things:

- an ODD : that is, a collection of TEI specifications
- a formal schema generated from that ODD, and its natural language documentation
- the set of documents considered valid by that schema

The TEI provides a completely unmodified schema called tei_all which contains all of the elements, classes, macros, etc. defined by the TEI. For all practical purposes a user of the TEI must make a selection from this cornucopia, and I will call that selection a 'TEI customization'. 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 unconsciously recalling distinctions proposed by David Birnbaum in a 2000 article (Birnbaum 2000, esp section 5.1) which talks of modifications as 'supersets' or 'subsets'.)

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'. Where this is not the case,

I propose the term 'TEI extension'. 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 the schema generated from it will consider valid is not a proper subset of the documents regarded as valid by the tei_all schema. Note that a change to the content model or the class memberships of existing TEI elements 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 constrains that order will be a TEI subset. 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'.

TEI extensions which include TEI elements or attributes whose properties or semantics have been significantly changed should place those elements or attributes in a different namespace. On the face of it, this means that any element containing such a redefined 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: TEI content models are defined not in terms of specific elements but in terms of references 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.

3 What is TEI conformance?

Umberto Eco remarks somewhere that a novel is a machine for generating interpretations. 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 explanatory force, so not all TEI customizations are of equal effectiveness or appropriateness. A customization documents a view of what it is meaningful to assert about a set of documents, specifically with reference to the existing range of concepts distinguished by the TEI. It does this by selecting the particular distinctions it wishes to make, possibly modifying some of them, possibly adding to them. I suggest that our assessment of the 'appropriateness' of a given customization – its conformance if you will – should take into account the way in which that customization is expressed.

There are good pragmatic grounds for wanting to know how a given customization has modified the TEI definitions. It enables us to make comparisons amongst 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 and others have pointed out, the pursuit of unmediated interoperability amongst TEI documents is largely chimerical, whereas the information provided by a TEI customization will often be all that is needed to make them interchangeable.

The notion of TEI conformance is introduced in Chapter 23 of the TEI Guidelines but the chapter falls short of providing a consistent formal definition, either of what conformance means, or how it should be assessed. One motivation for this paper is to start a discussion on how best to rectify that. I would like to conclude by suggesting that TEI conformance is more than a matter of validity against a schema. However, it should not be forgotten that there are still a few hard wired-rules built into the TEI model, which the customizer ignores at their (or rather, their potential audience's) peril.

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

 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>.

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 that once you have encountered another nested within it, only div elements at the same hierarchic level are permitted until it finishes ; this is not to say that a non-tessellating division element might not be useful, but if one is defined it most be distinguished clearly from the existing TEI , for example by placing it in a different namespace.

Breaking these rules may have unexpected consequences. For example, a customization which removes the element <title> will result in a schema in which no TEI Header 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.

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. However, in the case of a TEI extension, it is unreasonable to require that valid documents should also be valid against tei all. Validation of a document which uses a TEI extension can only properly be performed by a schema generated from the ODD defining that extension, and may additionally require the use of a namespace-aware validator such as onvdl [1]

This is one reason why validity against tei_all has limited significance in assessing the status of a customization, other than to determine whether it is a TEI subset or a TEI extension. The TEI was designed to facilitate either kind of customization, and either should be considered equally 'conformant', if that term is meant to imply something about coherence with the design goals or recommendations of the Initiative.

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 specialised 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.

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 memorably called the 'hellfire of ontology' into which the descriptive markup project has plunged an entire generation *[Note: (Raymond et al 1996)]* But I do not think it invalidates the general principle that TEI conformance entails a respect for the consensus, just as much as it facilitates autonomy.

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. In particular, I suggest, there are two important if largely unenforceable requirements of 'honesty' and 'explicitness'.

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'. If your 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 counter-productive. (Honestly made misreadings are of course entirely forgiveable: an encoding always asserts an interpretation, not the absolute truth of that interpretation)

By 'explicitness' I mean that all 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. 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.

In conclusion, I suggest that we should say of a document that it is 'TEI conformant' iff :

- 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.

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 personalised TEI encoding can be simply down-translated to other, possibly less expressive, semi-standardized 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.

[1] Birnbaum, David J. *The Relationship Between General and Specific DTDs: Criticizing TEI Critical Editions* in *Markup Languages: Theory and Practice,* Volume 3 Issue 1, December 2000, Pages 17-53); online at `http://www.obdurodon.org/djb/tei-crit/`

[2] Burnard, Lou and Sebastian Rahtz: *RelaxNG with Son of ODD.* Extreme Markup Languages.

[3] Burnard, Lou *Resolving the Durand Conundrum*, *Journal of the Text Encoding Initiative*[Online], Issue 6 | December 2013, URL : http://jtei.revues.org/842 ; DOI : 10.4000/jtei.842

[4] Holmes, Martin *Whatever happened to interchange?* in *Digital Scholarship in the Humanities*, 2016.

[5] Darrell Raymond, Frank Tompa and Derick Wood. 'From Data Representation to Data Model: Meta-Semantic Issues in the Evolution of SGML.' in Computer Standards & Interfaces 18 (1996): 25-36. doi:10.1016/0920-5489(96)00033-5

[6] Sperberg-McQueen, C.M. and Lou Burnard *Design principles for text encoding guidelines* (TEI ED P1, 1988, revised 1990)

Notes

[1]onvdl routes different parts of an XML document for validation against possibly many different schemas, using the Namespace-based Validation Despatching Language, defined as part 4 of ISO 19757