2013

Development Principles for Virtual Archives and Editions

Peter Shillingsburg
Peter L Shillingsburg, peter.shillingsburg@gmail.com

For Variants: The Journal of the European Society for Textual Scholarship

Peter Shillingsburg,
Loyola University Chicago

Based on Paper for ESTS 2012, Amsterdam

Development Principles for Virtual Archives and Editions

Scholarly editing has been a leader in the use of computing in the humanities and yet many of its tools are borrowed or adapted from applications that were designed for other purposes. This is good to the extent that adaptations built on the shoulders of previous efforts have saved time and provided desired results, but it is bad when off-the-shelf solutions result in compromises, not exactly what the scholarly editor wanted, even though they might help a project meet a funding deadline. The call for tools designed and developed for specific purposes in scholarly editing has been already made by many. Interesting projects are underway: AustESE (Australian Electronic Scholarly Editing) is aggregating and integrating a series of tools for scholarly editing (http://itee.uq.edu.au/~eresearch/projects/austese//); TextGrid, which unites a number of European projects (http://www.textgrid.de/); and InterEdition (http://www.interedition.eu/) has generated a great deal of energy around the development of tools and environments for editions. However, while varieties of interesting projects sporting new tools and designs are necessary to explore the options and possibilities that can be developed, we lack a comprehensive and thoughtful set of principles to define the characteristics that our tools, content, displays, and environments do or should incorporate.

It may still be too early to do this, but I wish to propose a beginning for such principles. Humanities Research Infrastructure and Tools (HRIT, pronounced Writ) is not set of tools but rather a set of principles for the construction of tools and environments for developing, maintaining, and publishing scholarly textual archives / editions / commentary and pedagogical presentations. I am keenly aware that these principles have evolved out of both the specific experiences of the team now working at Loyola University Chicago[1] as well as from our acquaintance with many projects being conducted elsewhere. That is, some of these principles are already shared by many projects, though I have not seen them articulated in any organized fashion.

HRIT principles call for a content management system or framework within which to offer tools for creating digital surrogates for physical text archives and new editions, and that can store and display texts in combination with contextual, analytical and critical content, and a system for displaying and navigating the materials--which all together I like to call Knowledge Sites.[2] I stress the integrated, but modular, system, because doing so has a significant impact on how one views the individual components of a Knowledge Site. Looking at components (images, transcriptions, commentary and/or tools) in isolation often leads to digital solutions that are too local. At the same time, some clumsy monolithic solutions of the past lead us away from over-integration and over-generalizing. It is useful to think of the

---

[1] https://sites.google.com/a/ctsdh.luc.edu/hrit-intranet/

[2] I first proposed "knowledge sites" in *From Gutenberg to Google: Electronic Representations of Literary Texts* (Cambridge UP, 2006), ch 4, where I tried to outline goals and ambitions for digital editions, but I was not at that time prepared to make useful principles for the development of the technical aspects for knowledge sites. Instead I proposed a list of twenty-one capabilities and content that one might desire in a knowledge site (91-92).

needs of specific projects--specific content-driven textual events--rather than trying to anticipate every need for every project. Modular construction and flexible user-adaptability are two essential elements of desirable systems.

Implementation of HRIT principles at the Center for Textual Studies and Digital Humanities, Loyola University Chicago, began with a content management system, in our case called "Mojulem," developed by our chief technical officer, Dr. Nicholas Hayward. There are other CMSs available but none, that I know, designed from the ground up to present knowledge sites for textual scholarship. Our project-development tool-set is still incomplete--will probably always be incomplete or at least be continually evolving. Our content is limited, so far, to a few projects, the most advanced and complex at the moment being the Woolf Online project for *To the Lighthouse*.[3] We used that project as the prime testing site for the principles enunciated here. And our recommendations for quality control in methods and procedures are also evolving as we continue to learn from mistakes. But HRIT principles for tools and environment do not dictate which content management system to use, nor which encoding system to use, nor which set of tools to use. Instead, they identify the kinds of functions that should be supported and point out pitfalls of solutions to be avoided.

Four Basic Principles

1st Principle

For literary works created in pre-digital eras, manuscripts and print primary materials are physical. The digital surrogates representing the primary materials are secondary--a first level of interpretation. The critical analysis of the materials is tertiary--another level of interpretation.

Primary	**Secondary**	**Tertiary**
Physical Origins: Manuscripts and Printed Documents	Digital Representations: Surrogates for Physical material	Scholarly and Critical Analysis, Annotation, and Commentary

HRIT's basic humanities research concept is that for verbal works originally inscribed as physical manuscripts and print documents the ultimate primary resources for textual scholarship are physical forms in physical archives, often unfortunately distributed in more than one place. It follows that digital surrogates are secondary. Electronic archives, for all but born-digital materials, stand at an interpretive remove from the primary materials.. Their appeal is never that they are fundamental; instead, it is that they are accessible anywhere, malleable, searchable, and capable of being analyzed and commented upon. developing a digital surrogate involves a modicum of interpretation and is therefore "secondary" not "primary", but the level of interpretive input by the creators of the digital surrogate is (should be?) qualitatively and categorically different from the interpretation involved in annotations, links, and commentary. In some ways, the secondary surrogate digital archive is actually preferable to the primary physical one, but one can/should never forget that it is secondary and capable of misrepresenting the "real thing."

[3] Launch date for www.Woolfonline.com is 1 October 2013, a URL originally used for the pilot project.

2nd Principle

Digital imaging is the closest one can come electronically to the physical primary original. Digital transcriptions, by contrast, are new typesettings, regardless of their mode of creation.

Images:	**Transcriptions:**
Iconic representations	Lexical representations (i.e. reprints, new editions)

In the world of surrogate digital archives, *digital images* are as close as one can come to (unmediated?) representation of primary materials. Digital images capture iconic values accurately enough (sometimes more clearly than the originals), albeit in only two dimensions, without weight, texture, substance, smell, or portability.[4] By contrast to digital images, *transcriptions* have the same relationship to primary documents as do reprints and new editions, which are re-settings of "type" (to use an archaic word in its generally accepted new digital meaning). Transcriptions are new editions (bibliographically speaking) requiring transformation of the semantic symbols and arrangements of an original into a new medium--for which there is no method that automatically ensures accuracy. Every transcription retains a potential for misrepresenting the lexical content of the originals, and transcriptions are stuck with the inescapable certainty of failing to represent the iconic values of their source documents.

This second principle, about the nature of digital images and digital transcriptions, is important to stress because so many Internet sites treat transcriptions as if they were the work itself rather than a new edition. A transcription created in 2003 of the text of an 1833 work is a new, 2003, creation; it is not an 1833 text even if the source of the transcription was an 1833 copy of the work. An important corollary to the distinction between physical originals and digital representations of them is the fact that a digital representation, whether image or transcription, is a representation of a single particular physical document. A document is just one item in an array of variant items, each representing the work differently. One physical document represented in image and transcription, represents an item, not the work--just a copy of the work.[5] One cannot represent a "work" digitally without including all the relevant variant forms that together represent the work. Bibliographers and textual critics traditionally have been very careful to identify which particular copy of the first state of the first printing of the work was used as copy-text for their new printed scholarly edition. In the digital world, students and scholars often seem willing to mount a digital image or a transcription on the Web and label it, for example, *The House of the Seven Gables* (1851) as if there were only one text of that work and as if all 1851 copies were identical and that this digital representation captured that mythical uniformity.

---

[4] Of course, digital projects are portable, but the word has a different meaning for digital objects from that which is applied to books and manuscripts. In some ways the digital is more portable. Perhaps we could use a distinction between substantial and insubstantial portability--though that is unsatisfactory for different reasons.

[5] The terms 'work' and 'item' are technical terms in FRBR (Functional Requirements for Bibliographic Records) guidelines for library cataloguing. Work is a category to which belong all the Items (individual physical copies) which can be further categorized as representing particular expressions of a work or a variety of manifestations of an expression. Thus, the digital form of a book is just one item of the many variant items that represent the work. An item is not "the work." Descriptive bibliographers and scholarly editors fully understand the relationship between a single copy and the range of variant forms that make up printings and editions.

Identifying in this way the relationship between the material archive (primary) and it surrogate digital counterpart (secondary) has the effect of turning scholarship and criticism--the explanatory results of analysis and interpretation--into tertiary material, one further interpretive remove from the originals. That is to say that the traditional distinction between original documents and analytical commentary is usefully pushed off one more step if we are to consider the electronic surrogate, itself a secondary or mediated construct, to be the object that is being analyzed.

3rd Principle

Modular (component) design and structure are best for digital tools, tasks, and content types.

Modular Tools **Modular Tasks** **Modular Content**	**Separating --->**	**1. Images** **2. Texts** **3. Tags and commentary**

HRIT principles reflect a growing trend in digital projects, committing to modular tools and modular storage of content. Images, of course, are separate, but the traditional mixing of texts and commentary in the same file (text with embedded markup and annotation) has very serious unintended consequences. A difficult but necessary principle for future development is that the digital representation of texts as transcriptions must not be mixed in the same files with the explanatory, analytical, and interpretive commentary. The reasons for separating these normally mixed elements need articulation, for the tradition of embedding markup in texts is strong. Intuitively we think of text and encoded enhancements as integrated, perhaps because when we invoke a database for images, transcriptions, and information about a text we want an appropriate integration in the rendered display on the screen. For several reasons, it is a mistake, however, to store these elements mixed together in a file. Very significant pitfalls can be avoided and advantages gained by maintaining archive content (images, transcriptions, and commentary) in separate storage units.

Just as it was difficult to use the terms works, documents, texts, and editing without confusion--because those words are often used interchangeably--so, too, it is difficult to write precisely about “markup” and “annotation” for text files because of different assumptions digital scholars apply to those terms. Exacerbating the problem is the multiplicity of purposes of markup to indicate how the text looks (special characters, font, format), to explain how the different formats function (the meaning of italics, bold, indentation, or extra spacing), and to include other analytical enhancements (commentary, explanations, and links). Furthermore, linguists will mark phonetic features and cultural historians will mark regional features. Two serious questions about text files that embed coding for any of these things are: what happens when someone wishes to add new coding to an existing file? and what happens when one wants to use a heavily encoded file for text analysis, linguistic analysis, or text collation? In the first case, the file is opened with some kind of text processor to add new coding--a process that invariably makes the integrity or accuracy of the file vulnerable to inadvertent change (or the owner of the files refuses permission for anyone else to add code, generously suggesting instead that such code be added to a copy of the file, thus, proliferating projects on the same text). In the second case, the user of the file will have to strip all the codes in order to have a “text only” file for analysis or collation--except that one will then have an unacceptably stipped down text that does not fully represent the textual features of the original that one might want to see in the analysis or collation. In short, embedding code in text files

is a bad idea--regardless of how understandable and useful it has been historically.

The separation of text from analytical and interpretive markup and annotation requires further examination. To begin, one needs to define “text only” by some other means than "that which can be typed with one stroke at a keyboard in ascii or unicode"--unless the result is a single unicode character that, like qwerty, is part of the text. Drawing the line demarcating “text only” by the mechanical capabilities of a keyboard saddles sentient, analytical humans with a definition imposed by the limitations of the software and hardware at our disposal. (I will tease out, in the next paragraph, the issue of who or what gets to draw that line between “text only” and interpretive analysis, here here the focus is on the what might constitute the difference between these two concepts.) When, instead, “text only” is defined by what humans find to be “textual” (i.e., semantically significant visible features in the physical written and printed documents), then “text only” includes everything about the sign system and its deployment that strikes the eye of the reader as having semantic force: e.g., letters, punctuation, diacritical marks, and other printed symbols, plus the meaningful deployment of white space through indentation, extra spacing, line breaks in poetry, plus certain kinds of type-font changes like the use of italics, bold, bold italics, special characters such as digraphs, and accented letters regardless of how many strokes on the keyboard it takes to render them digitally. For manuscripts, the list of visual elements with semantic force is somewhat longer, with strikethroughs, overwriting, insertions above and below lines and in margins, changes in writing instruments,and changes in handwriting (was it Cervantes or Menard), but not rips or coffee stains which are not textual, though they might affect text. These aspects of “text only” have two qualities: they register immediately to the eye, and they signify something about the text which, if the text of another copy of the same work did not have it, we would say there was a textual variant. Encoding systems already include ways to record these aspects of texts. That is not the problem. But when such basic representational encoding is mixed with other additional types of encoding, the resulting files immediately become unwieldy for any use other than that imagined by the first transcriber. If such files were offered for collaborative enhancement, they inevitably will incorporate new errors added inadvertently along with the new coding. Embedding code beyond that required for “text only” is the enemy of collaboration.

The line between text-only and interpretive code has been shifing. Ascii and qwerty drew the line first, one by having only 256 characters with which to work; the other by having only the keys on the typing keyboard, plus the shift key. It soon also had the control key but its implementation initially was so proprietary that it created serious problems for using the resulting texts. Now with Unicode there are many more single-character options for recording text-only, even if to gain them one might need more than one keystroke. Presumably the barrier line between what is “stuck-on code” and what is “inherent textual character” that is drawn by technology will get closer to the line that a humanist reading text would draw. But from a humanist point of view, that is the line the matters. First, I want a conceptual understanding of the difference between “representing visual clues to textuality” and “understanding what those clues mean”. The qwerty keyboard’s first achievement for most textual features is to represent them without comment: e is e and i is i. Encoding systems (including TEI) tend instead to want to represent what the clues mean: slanted letters mean a ship’s name or emphasis or slang or a foreign word. In fact, visually, all it is is slanted or italic type. Text only would indicate an indentation of x spaces; code would call it paragraph. The conceptual difference between what it looks like and what it means is important NOT because it takes fewer keystrokes (it frequently does not) but because coding “what it means” or “how it functions” is hugely more limiting than indicating “what it looks like”. To a

linguist, a bibliographer, a typologist, a stylist, a statistician, a capital letter is a capital letter, and an indentation is an indentation regardless of how each of these would encode its meaning. Text only would serve them each equally; encoding for one purpose will stymie the use of the text for other purposes. Text-only representation in the text-only file and associating all other coding by some stand-off means will make it possible for all users to attach their particular codes without interfering with another person's use of the text file.

Text collation is a particularly telling test of this principle. Collation programs should be able to detect and report all textual variation. Non-textual encoding, being non-textual, should not show up in collation. To facilitate text collation, encoded material embedded in text files is removed--including the encoding of the textual features just described. The resulting collations miss out important textual variation such as italics, special characters, accents, digraphs, and some special deployment of white space, and any other elements that require more than one key-stroke (encoding) to represent. Collation of texts stripped of all coding is incomplete, inadequate, and unacceptable--all because of indiscriminate embedding of non-textual markup and annotation. Just to clarify, by non-textual I mean any kind of explanation of text.. For example, italics is a textual feature, but identifying a particular use or function of italics, such as for emphasis, or for a book's title, is explanatory, not textual.

Encoding functions, uses, explanations, links, phonetic features, and endless other aspects of textual, linguistic, historical, and literary analysis is important, but embedding such in a text file severely limits--in fact prohibits--the repurposing and flexibility of the text file. Such encoding should be kept out of the text file in some form of standoff or database structure.

Separating specific visual features of a document that are deemed to carry semantic force (and therefore are part of the text) from all other encoding of functions and analysis (which are added critical enhancements not part of the text) is necessary but not simple, because what is deemed "text only" may depend on the period or the author or the purpose for which a transcription is being created. The fact that some textual feature cannot be represented by a single stroke on a keyboard, or by a single Unicode character in the text, and therefore requires markup, does not distinguish that textual element from text; it too is text. But because texts of different periods or produced for different purposes have potentially different requirements, each project creator will have to decide where to draw the line between what is essentially text and what is analytical or explanatory.

Furthermore, because TEI (Text Encoding Initiative) was designed to interpret function rather than format--generic structures rather than physical or visual ones--it is taken for granted by many encoders that the object of encoding it to indicate function and not to indicate format. Of course, one should encode function, but not in the text file itself. To do so limits the use of the text file in exactly the same way that embedding coding for phonetic analysis or identifying analytical elements of prepositional phrases would interfere with the use of such files for collation or stylistic analysis. I contend that to make a transcription easily repurposed, one must NOT encode function or analysis IN the text file. One must embed only the visual elements of the page that have potential semantic force; "text only" means what you can do with a single stroke of the keyboard plus everything you have to add to represent the visually palpable aspects of the source document that carry potential semantic force. For example, a word in italics should be coded IN the text as italic; any additional coding explaining that the italics means emphasis or a book title, or a ship's name belongs OUT of the text. The primary task of transcription is to record the aspects of the text that have potential semantic force; it is not to explain what that semantic force is supposed to mean. Embedding editorial impulses into the text is no service to

persons wishing to use the text for a different purpose.[6]

Of course, one purpose for transcriptions is so that interpretive, functional, generic, explanatory enhancements can be added, but these editorial, analytical, and critical tasks should not impinge themselves in the text file; instead they should be kept separate, pointing to the text file and rendered by a browser on the screen, appropriately integrating text and commentary there. Furthermore, there is no reason to embed interpretive markup in text files, for there are perfectly functioning standoff modes for such markup/annotation/linking enhancements. A transcription that distinguish between the *fact* of italics from the *meaning* of italics--the former being textual the latter being interpretive--is a transcription that offers itself usefully for collation and for endless amounts of standoff encoding of analysis and commentary that protects the text from inadvertent alterations. The optimal scholarly editing tool, designed for the purpose, is one that embeds code for textual facts and stores in standoff mode all other markup and annotation. No editor should leave these distinctions to the limitations of ascii or keyboards.

Besides HRIT, the AustESE project is the only project I know that distinguishes textual markup, which it calls "Properties," from "Annotation" markup, which includes all analytical and critical enhancements. Interestingly, in that project even textual properties are treated in standoff mode, though I think that embedding properties (minimal textual code) in the text file respects their standing as essentially textual elements and, perhaps more practically, ensures that they are in the files being compared by collation programs.[7] Alternatively, textual properties could be stored in standoff mode if they were strictly isolated from other markup--perhaps being treated as a special category of default markup, while other markup is offered in a list of choices that users could invoke if they wished. That way, a sophisticated collation program (which does not yet exist) could invoke the minimal markup, and include it in the comparison of texts.[8]

As a side note, one could hope that in a sophisticated collation program, such minimal textual markup (normally indicated as a range marked at the beginning and end) would be converted to character-level before collation. The reason is that range markup appears in collation to apply only to the beginning and end of the range, whereas in fact, it applies to every character between the beginning and end of the range. If a variant occurs within the range, collation will not find it unless the code is applied at character level. Character-level markup would be practically unreadable by humans, of course, so the results of collation will need to be rendered in readable form for screen displays and printouts. No human would do the encoding at character level or ever see the text so encoded. In the following example collation reveals the variation between "white" and "black" but would at present miss that one is in italic and the other is in roman because the variant is buried in the middle of a sentence that is all italic on one version and all roman in another:

Her tanned face turned white at the sound.

---

[6] The phrase "editorial impulse" as opposed to "archival impulse" was coined by Paul Eggert at a conference. I have not yet seen it in print.

[7] The practice of embedding markup in text files and then extracting the text from the markup for collation defeats both efficiency and accuracy unless some method is used to retain "minimal textual properties" in the extracted text. It is more efficient to make the distinction between text (including properties) and annotation markup a fundamental one that separates these essentially different coding categories. But that is not enough for, range markup of textual properties would still create anomalies in collation that are also unacceptable. This problem can be addressed with character-level markup.

[8] Yes, I did mean that CASE, CollateX, and JUXTA are unsophisticated.

Her tanned face turned black at the sound.

I want a collation program that, in addition to finding that one says white and the other says black but will also say one is italic the other not. What I get now is somthing like

Her vs Her
white vs black
sound vs sound.

Not good enough.We do not have such collators yet, but perhaps a legitimate human need will lead someone to create a useful tool.

It has become customary to say that transcription itself is interpretive, that it already introduces an editor's critical thinking into the facticity of transcription--i.e., text only transcription is already coded. And while that is palpably true, especially when originals are ambiguous, it is a different sort of interpretation, categorically, from the explanation of functions, meanings, or contexts that editors also add to texts through encoding. Any markup beyond minimal textual properties, "text only", constitutes enhancement and explanation of a different order, even when it seems necessary to make the visual element understandable. We may, for example, think it important to indicate the order in which we think changes were made. But that explanation is extra-textual--not part of the text itself. Even if a comment is helpful, like that the italics was added by a compositor or that the italics indicate a ship's name, these too are extra-textual--desirable and helpful, but to be stored in stand-off mode, not embedded in the text. Even calling attention to an ambiguous mark in the text, which the editor has interpreted in one way, is material that should go into standoff mode.

To this point I have focused on the usefulness of standoff markup, implementing the principle of component structure to enhance repurposing of text files and enabling the mutual use of a stable text file for a variety of different users--enhancing collaboration. But there is more to be considered.

4th Principle

Distinguishing textual properties from analytical and explanatory markup is essential to the durability of archival surrogates.

Minimal Markup (embedded)		**Enhancement Markup (standoff)**
Visual aspects of material texts with semantic force such as indentations, line breaks in poetry, italics, special characters, etc.		Analytical and Explanatory markup – added critical value such as annotations, identification of functions for italics, identification of linguistic and lexical anomalies, cross references, links
Text is Finite (Correct it; Close it) Stored separately as	**Analysis and criticism is Infinite** (Open it; Leave it open)	

foundation	Stored as standoff , extendible components

That markup of all kinds is commonly embedded in text files may relate to the fact that, in the first decades of web-based projects, digital *transcriptions* were designed to stand alone as representations of the primary material objects. Digital *images*, up until the first three or four years of this century, were too expensive or loaded too slowly to be considered standard equipment for digital projects. It became natural to embed in the text coding for both markup and annotation. Just as in print editions, there seemed to be no reason to make a clear distinction between the archival and editorial impulses. In the present era, however, images are feasible, and, I contend that for archival purposes, images are primary--the closest one can get, digitally, to the physical originals. Transcriptions, though absolutely necessary for collation, stylistic and other linguistic analysis and for most forms of repurposing, stand at a further remove from the originals. For the virtual archive, images are first and foremost, providing verification for the transcriptions necessary for most computer-aided analysis. Transcription and coding may help a person understand what one is looking at, but seeing is more direct and verifiable than explanation. Seeing in the image that an addition to a manuscript or proof or a marginal comment is in pencil or ink builds more confidence in a user than being told that it is so. Users like the convenience of transcriptions to help decipher a difficult image; transcription is a crutch and a convenience but the image gives a better sense of the evidence. Even if one needs a transcription in order to search a text, one still need not see the transcription, which can be hidden behind the images such that it almost appears that one is searching the image, although the software is really searching the background transcription. Searching the transcriptions but seeing the image of the original is a form of verification.

Beyond the fact that we no longer need to embed non-textual enhancement markup in text, there are simple reasons for preferring the separation of text from analytical and explanatory markup. First, text and its minimal representational markup or properties is finite, whereas markup for analytical and explanatory enhancement is endless. Once all the relevant texts have been transcribed and proofread and brought to the highest level of accuracy, and mapped onto their respective images, the door should be closed to further interference with the text files. That is, the surrogate archive is now ready to serve as a resource, a foundation for critical enhancement built ON it, not IN it. The only thing that would make any textual scholar wish to reopen a text file is the discovery of a new authoritative witness to the work or the discovery of error in the original work.

One could also, but one should not, close the door on extending analytical and explanatory enhancement markup for a knowledge site. Once finished and locked, the surrogate archival materials should be a stable core or foundation upon which scholars everywhere should be able to build commentary and explanation and critical opinion.[9] Unfortunately, closing projects to further

[9] A useful analogy might be drawn from online computer games, where the core programs and materials and images for use in the game are protected from damage by any player but available for players to enhance and build on in almost unlimited ways which demand their attention but leave no marks on the core capabilities of the game. Likewise, archival textual materials are the givens, the core; they are not manipulable; but they should be available as the base for enhancement and project building, not only to the project developers but to persons whose methods

commentary is normal practice with projects that embed enhancement markup in text files. That is because embedding growing amounts of analytical and enhancement markup in transcription text files leads to files of unwieldy length, vulnerable at each opening to inadvertent corruption, consequently becoming increasingly worse proofreading nightmares, and because embedded code tends to restrict the repurposing of text files. Whenever I ask project leaders if I can please add my coding to their texts in their projects, they always say NO. They might offer me a copy of their texts so that I can add my analysis to it in a separate project, but then a user of both projects would have to access our projects separately. That is not collaboration.

It is counterproductive to think of the transcription of a text in isolation from the array of alter-texts and tools required to make a digital surrogate archive. For example, if one says that a text with any embedded code is already interpretive and will therefore not serve everyone's needs, one is already taking both a too general and too isolated view. The problem arises from trying to think of "everyone" in relation to "all texts." A text of Virginia Woolf's *To the Lighthouse* is not going to have to meet the transcription standards or use the conventions developed for medieval texts--marking up the textual properties appropriate to Woolf's text is not going to ruin the text for medievalists or classicists because they will not be using the text of *To the Lighthouse*, nor will modernists be using the texts of medievalists and classicists--not, that is, without changing hats and becoming the sort of scholar who uses those texts. The point is that the conventions for texts in a field of study should worry first and foremost about the conventions in its own field and not worry too much about universal interoperability outside home territory, outside the field of texts that have enough in common to be considered a field of study. That might be nice, but for whom is that nearly impossible goal designed? Restricting embedded coding to minimal textual elements is already the step that needs to be taken to make transcriptions flexible. Being flexible is good, but being appropriate to the task at hand is better. The assumptions about the nature of texts among Woolf scholars will have a great deal in common with those held by modernists--and probably also with scholars interested in verbal texts of the last three centuries. Embedding minimal textual property markup will enhance, not destroy, the texts for those groups. As for the goal of interoperability itself--creating texts that can be used across disciplines, not just across platforms--it remains an unachieved goal.

Likewise, it is a mistake to think of a transcription in isolation from the image of the document or in isolation from a browser capable of rendering for human reading both standoff markup and character-level markup prepared for computer processing. Thinking of the transcription in the context of a knowledge site, with multiple relations to other texts and capable of use with multiple tools for rendering complex underlying computer files as user-friendly reading forms reveals how desirable it is to remove the formidable difficulties humanists have in dealing directly with marked-up texts. I dare say many digital humanists shudder now at the idea of using command-line programs. So, too, should they shudder at *seeing* a text with embedded code. This point is especially important when considering the advantages of character-level markup. Computers must see texts with code, but humans should see texts without code--with code rendered for human consumption. That includes textual scholars who wish to build textual projects but who haven't the time to become computer scientists or digital humanists. The goal is to build tools that will let them see the results of their actions without having to

---

and goals differ from those of the original developers.

decode for themselves--an activity they should be able to choose to do; not one they have no choice about.

HRIT recommends the use of character-level markup in the place of markup indicated as ranges that must be nested for another reason, also. It eliminates structure and the overlapping hierarchies problems endemic to XML; incidentally, but trivially, it also renders the markup unsightly to humans, but humans need never look at it. The usefulness of character-level markup and standoff modes can be appreciated by humanists, who should not have to understand the technical explanations any more than they need to understand how Windows and Google Drive and spreadsheets work. The primary points here are a) the principle of modular structure and modular development, b) methods that ensure the integrity of basic textual scholarship while opening the way (separately) for long-term dynamic growth through collaborative scholarly and critical enhancement, and c) the principles that encourage tools designed specifically for scholarly editors of literary texts.

Other Considerations

There are a few other matters to mention because, though important, they do not need elaboration here. They include the metadata that identifies sources, describes methods, and gives credit to developers. They include principles for markup which TEI has developed to a fine pitch of nuance. Whether embedded or in standoff mode, TEI will be with us until something better is created, and even then, the thinking that has gone into TEI's analysis of textual features and needs will survive. The fact that HRIT principles do not require that markup be TEI conformant or even that XML be involved at all is beside the point since HRIT principles uphold the idea that any tool and storage system that does NOT use TEI / XML should have a system that accomplishes the same objectives at at least the same level of granularity. Furthermore, such a new system should have a mechanism for importing content from and exporting content to these almost universally used standard systems. HRIT does not object to TEI as a marking up system--in fact supports TEI as a sophisticated tool for enhancing textual projects; but it seriously objects to embedded non-textual enhancement markup in text files representing primary material. And it objects to the limitations of current XML with its nested, hierarchical, range markup system. Separating text-only from enhancement makes both transcription of text and enhancement of text more versatile and flexible.

HRIT principles also call for modularity in a content management system to provide an environment that supports multiple tools for multiple functions, including optional tools for doing the same work, for replacement of tools, and for substitution or addition of content without affecting the functionality of other parts of the knowledge site. It also calls for modularity in the tasks of project development. For example, first create the textual surrogate and, only after that is finished, begin the task of annotation and enhancement markup, using an editor designed for standoff, character-level markup.[10] And for another example of modular methods, use your collation programs first to help proofread and, then, when your standard of accuracy has been reached for the texts, use collation again to identify and store actual textual differences. And having identified textual difference, then ask that the display mechanisms offer a variety of ways to see textual difference; the default output of a collation program should not determine the display options. That is at least three separate tasks, not one

[10] There are two such editors under development at the Center for Textual Studies and Digital Humanities at Loyola Univ., but my point is not that project developers should use a particular editor. Instead, I've pointed out adequate reasons not to embed markup and therefore our community must explore and create alternatives. Not to do so is like finding that your otherwise very good car is stuck in a ditch and deciding that that is okay.

extended one.

In the face of tedious transcription tasks, many of us rely more than we really want to on OCR production (unsatisfactory OCR usually, more's the pity) or we send images to some country where low wages for skilled workers makes it feasible to have texts typed or even double-keyed. Or, if we type the texts ourselves, we tend to want to jump the gun by annotating and marking up the text as we work on getting the text accurate. But that IS jumping the gun because the value of any textual project is undermined by every error that remains in the text or is introduced inadvertently while adding new markup. Clean it first, then annotate it--in standoff mode--that is, in an environment that protects the core textual data.