



Gabrielle Kirilloff is a senior English Literature major at the University of Pittsburgh. She will also receive a Certificate in Russian and Eastern European Studies. Her interests include comparative literature, the history of the book, gender studies, digital humanities, and media studies. Currently, she is a TA for the English Literature class, “Computational Methods in the Humanities” and is pursuing several digitally oriented humanities research projects. She hopes to pursue a PhD in English Literature and eventually teach at a university. Information about her research can be found through her homepage: <http://gk.obdurodon.org/>

FORBES & FIFTH



<Traversing_the_Tree/>

Exploring Ontological and Pragmatic Approaches to Markup Theory in Digital Humanities

Introduction

Digital humanities, the area of study that explores the intersection of computing and the humanities, not only affects the presentation and accessibility of literary resources but also creates opportunities for scholars to ask new questions. As computational tools — particularly those used for digital transcription — become more prevalent in the humanities, it is imperative that scholars assess the ideologies behind the technologies that have been appropriated for academic use. Many computational tools used for literary research treat texts as an aggregation of specific formal features. There is an important distinction that needs to be made regarding whether these features reflect the true meaning of a text or whether they simply allow for one practical approach to text encoding. The reason that an exploration of these approaches is necessary is that the methodologies used in digital humanities have the power to limit or expand the potential of emerging humanities scholarship.

The integration of computing tools into literary research is shaping the way scholars can interrogate their texts. This is not to say that the study of digital humanities is making literary research more objective. The anxiety that computing tools are taking the “human” out of the humanities overlooks the fact that computers do not have the power to analyze or interpret data. Thomas Rommel, a literary scholar and digital humanist, astutely points out in “A Companion to Digital Humanities” that the strength of using computing tools in literary scholarship is that they provide speed, accuracy, unlimited memory, and instantaneous access to virtually all textual features, but are still completely reliant on the scholar.¹ The systematic analysis of literature based on quantitative textual features has always been a part of literary study. For example, scholars have used concordances for centuries to study patterns and data sets present in a text. Digital tools allow scholars to sample larger amounts of texts and perform a more complete comparison of the differences among sets of texts. If anything, with more data readily available

at their fingertips, scholars have to work harder to figure out which lines of inquiry will yield productive results.

Understanding these emerging relationships involves investigating how digital texts themselves are created, a process at the heart of all digital humanities scholarship. The idea that scholars should interrogate the procedures behind the creation of a text is not new; the study of how transcription methods and editing practices affect print texts is well established. The encoding process demands that the scholar makes a decision in terms of classifying elements of the text. Similarly, when new versions of a print text are created, editorial decisions are often made about ambiguous elements of the text. For example, the punctuation used in different print versions of William Blake's *Songs of Innocence and Experience* varies because it is impossible to establish all of the punctuation in the illuminated versions. This is both because some of the original punctuation is unclear, and also because some of it varies across the different illuminated manuscripts.²

There are also differences between print and digital text creation. Perhaps the most significant is that, even if a transcribed digital copy of a text is formatted in the same way as the print version, acts of interpretation and classification occur during the encoding process and exist in the marked up version. This is because the process of markup used in digital humanities is intended to be descriptive.³

Descriptive markup seeks to ascribe meaning to a text through the categorization of textual features. Even though markup is not visible in the readable version of the text, it is still a necessary part of the creation of the text. This is not the case in the creation of a printed text. If the goal of creating a print transcription is to visually reproduce the text, classifications of meaning do not have to be made. In such a case, whether or not the mark at the end of a poem was intended as a period or a comma is irrelevant, but its shape, color, and size are important.

In the creation of a digital version of a text, however, decisions about classification are made, even if the focus is on rendering a digital version that is visually similar to the original. In addition, the marked up version of a text exists for as long as the "readable" version exists. As a related point, different modes of classification and textual interpretation in the creation of a digital text can be made to achieve the same visual result. For example, if two scholars were to classify and describe the


same block of text differently, as a paragraph and as an epigraph, the text could later be rendered to look the same way visually regardless of the classification.

The Implications of Encoding

If it is true that the interpretive decisions made “behind the scenes” do not necessarily affect the presentation of the text, why do the methodologies used for encoding matter? The answer to this question has three components. The first is that, while descriptive markup may not affect the potential visual presentation of the text, it does significantly affect any queries run on the text. For example, suppose scholars encounter a text in which the content between quotation marks includes words that are not part of a spoken quotation: “Hello, said the girl, how are you?”

Assuming that the scholars do not want to compromise the integrity of the text by relocating the question marks, they are faced with a number of ways to classify and interpret this text: they could label the example as either one or two separate quotations. Whether or not the scholars decide to label this instance as one quotation or two quotations separated by the phrase “said the girl” does not impact the way that the text can be rendered after the markup process. However, the methodology would affect a count of quotations within the text and would also impact the number of letters contained within the quotation. For example, in part of a study that used computational tools to measure the frequency

of male and female speech in folk tales, these types of quotations were treated as one speech act. Since speech was measured as a type of agency, the scholar’s decision directly impacted the results of the research.⁴



“The creation of a digital text is an active, interpretive process.”

The second reason that the encoded text is important is that the creation of a digital text

is an active, interpretive process. This process is cyclical: on the one hand, the way that scholars perceive a text shapes the way in which they choose to encode the text. On the other hand, the constraints of the

markup process and the ideology behind it affect the way that scholars perceive a text.⁵ This last point is particularly relevant when discussing the importance of markup philosophy, since it acknowledges the potential for the system of formalizing a text to limit the literary scholar. It also suggests that the study of markup should be focused on the activity of encoding and how scholars interact with this process.

Building off of the relationship between markup and scholar, it becomes clear that the rules and constraints of the markup system used can impact the reader as well. The scholar's choices of categorization and formalization may impact a reader's interpretation of the text. For instance, if a reader were to observe that a line of a poem by John Donne had been encoded as an example of "metaphysicalDeceit,"⁶ their interpretation of the line might change in the same way it would from reading a piece of criticism.⁷ This makes sense, since both the markup process and literary criticism are based on interpretation. However, since the scholar's act of interpretation is in part based on the constraints of the markup system, the reader's perception of the text is indirectly influenced by the markup system.

This argument is dependent on the reader seeing and understanding the encoded text. Because most people who read digital texts do not observe them in their encoded form, this is a weak argument. However, it is likely that coding literacy will increase over the next several years, making it more probable that readers will view the source files of documents. The motivation behind this might come from a desire to "decode" the code or understand the methods of interpretation and inscription behind any given text, something that is present in the study and consumption of print media today. A more immediate and convincing argument about the relationship between reader and scholar is that the digital resources and information available to the general public are determined by the questions scholars ask of their texts. By way of a trickle-down effect then, the structure of the markup system also impacts readers even if they have not viewed a text in its encoded form.

In light of the relationships among text, markup, scholar, and reader, the next logical question is: what are the constraints of current markup procedure, and how are they affecting research? In order to fully discuss this question, it is necessary to give some background information about the history and evolution of text encoding.

Text Encoding

The most current and widely used method of describing, or marking up, a text so that it is machine-readable, is XML (Extensible Markup Language). XML grew out of SGML (Standard Generalized Markup Language), a language that was developed for data encoding and the sharing of machine-readable texts in the late 1980s. It is important to note that, although SGML was developed to encode texts, its original goals were very different from the way humanities scholars now use XML. SGML was intended to facilitate the sharing and storage of large-project documents in law, government, and industry.⁸ When SGML was developed, it was assumed that the system of formal features described would be based exclusively on a text's genre. SGML was not developed for application in humanities scholarship and the primary concerns driving its creation were practical. One such practical concern focused on creating a model that would be accessible and easy to use.⁹ The history and development of text encoding suggest that it might be necessary for literary scholars to reassess and reclaim the tools they are currently using. It seems strange that methodologies developed around principles of practicality are being used to determine textual interpretation and meaning. This begs the question: are the text encoding methodologies used in literary scholarship rooted in principles foreign to their current purpose and use?

Surprisingly, most of the scholarship on the philosophy of markup is from the 1980s and 90s and is thus focused on SGML as opposed to XML. The discontinuation of the SGML debate was not a result of significant improvements. Rather, it seems as if with the creation of XML, most scholars accepted the shortcomings of SGML and XML as unavoidable. Although there are differences between SGML and XML, they both share the same data modal. Encoding a text with XML involves wrapping the text in tags, markup that describes components of the text. Tags divide the text into elements, while elements include tags and everything in between them. For example, in the following sample, the text, "This is a sample," is wrapped between a sentence start tag and a sentence end tag. End tags always contain a slash. The entire example constitutes a sentence element:

`<sentence>This is a sample.</sentence>`

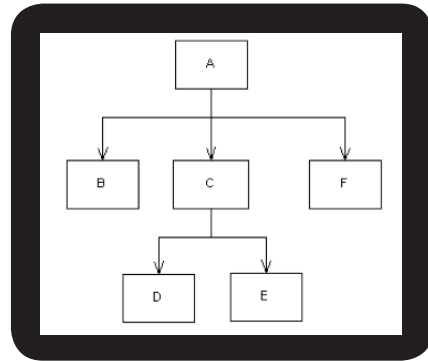
FORBES & FIFTH

XML, unlike most other languages, does not have predefined tag sets. This essentially means that the scholar determines the “name” of a tag. In the above example, the word “sentence” could be replaced with any other word and still be processed the same way. This aspect of XML makes it remarkably similar to a human language, and, in fact, the experience of writing XML is in many ways similar to writing prose. It is important, however, to remember that the “names” are not read or understood by the computer. Because humans can comprehend the words written in XML, it is easy to think that a computer is processing the text in a similar way. The reality is that the computer is processing XML as binary machine code.

Writing in a language that does not have pre-defined tag sets increases the effect of the markup process on the scholar. For example, take the following section of text: “...able wind. Considering the grand way these moments are photographed, it almost appears as if Navidson is trying through ever the most quotidian objects and events to evoke for us some senses of Holloway’s epic progress. That or participate in it. Perhaps even challenge it.”¹⁰ In its source document, Mark Danielewski’s *House of Leaves*, the above snippet of text is displayed with line breaks between a preceding and following paragraph. If the scholar were to encode this snippet of text with HTML, a language with predefined tag sets, a `<p>` tag, or paragraph tag, would most likely be used. There are no other options in HTML that come close to describing the above sample, but being given the complete freedom to “name” the sample makes the classification more difficult. It also makes it more thought provoking. Is it, in fact, a paragraph? The first word, which is part of “untouchable,” is cut off and there is no indentation. The scholar could choose to label the sample as any number of things, including `<paragraph>`, `<text_block>`, or `<incomplete_p>`. Each of these options represents a separate interpretation of the text.

Thus, the scholar is free to choose the words that describe the text, but SGML, XML, and the TEI (a standardized subset of XML that includes its own rules and list of valid textual components) have specific rules about the way in which the text can be tagged. All of these languages are based on the idea that texts can be represented as ordered hierarchies of content objects (OHCO). This approach is modeled after the computer science concept of a tree data structure. Tree structures are

based on a hierarchy in which individual components, or nodes, form parent, child, and sibling relationships. In the following diagram, the parent node is represented by box A. Boxes B, C, and F are siblings because they are the immediate children of A.



The rules of XML revolve around forming and maintaining this type of hierarchical structure. If the structure produced does not follow these rules, it is said to be not well formed and cannot be processed. In order for a document to be valid, it must have a root node and be properly nested. A root node is the top node in a tree and, consequently, is the only node not to have a parent. In this diagram the root node would be box A. In order for markup to be properly nested, there must be no overlapping hierarchies.¹² If the text is going to be used for research purpose, after it has been marked up, the hierarchical structure is used to traverse the document and pull out information relevant to a scholar's query.

Overall, the tree-based approach works very well. Having to impose a hierarchical structure often leads scholars to notice new or unusual structural features of a text. Perhaps this explains why in the last fifteen years XML has enjoyed such success, despite its various shortcomings. The main reason for the success of the OHCO based system is that most texts lend themselves well to a hierarchical structure. Books contain chapters that contain paragraphs that contain sentences, etc. However, this model is not foolproof. In particular, many texts contain overlapping hierarchies, textual features that do not nest within other features. An example of this would be continuous dialogue that extends over multiple paragraphs. Although there are a variety of methods in place to circumvent this issue, overlapping hierarchies need to be thought of as something more than an "exception" to the rule. The fact that they exist at all suggests a more fundamental problem with the way that hierarchy-based methods define a text.

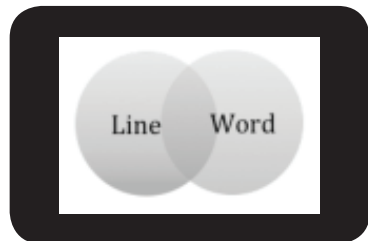
Overlapping Hierarchies

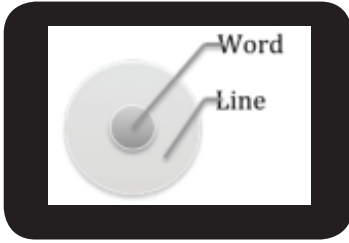
SGML was originally intended to represent a single, logical hierarchy in a document. It was assumed that each document could only have one logical hierarchy and that the hierarchy would be determined by genre. For instance, the typical document hierarchy for a script might be: cast list, performance history, title, stage directions, act, scene, line.¹³ Once SGML and XML started to be used for literary scholarship, it became obvious that there is no single “standard” hierarchy for any given text. There are multiple, perhaps even unlimited, combinations of potentially interesting features in a text, regardless of its genre. In one situation, rather than look at the acts and scenes in a script, a scholar might be interested in examining the length of speeches and the use of verbs. Hierarchy came to be dependent on interpretation; thus, markup reflects a theory of text, creating an unlimited number of logical hierarchies to be found and encoded in any document. This realization opened up the door for a number of problems.

Overlapping hierarchies can occur almost anywhere, including the following: when a paragraph extends over multiple pages, when a metaphor extends over multiple sentences, or when the hidden message becomes encoded in an acrostic. While overlapping hierarchies can occur in any genre of text, they are often found in poetry. This type of overlapping hierarchy is often caused by enjambment. This occurs in the poem “The Unrhymable Word: Orange” by Willard Espy: “The four eng-/ineers/Wore orange/brassieres.”¹⁴ If a scholar attempted to tag both the lines and the words in this poem, the first two lines would create an overlapping hierarchy:

```
<line>The four<word>eng-</line><line>ineers </word></line>.
```

The relationship between words and lines in this poem does not corresponded to a strictly parent, child, and sibling structure. The first diagram shows the relationships between lines and words that exist in the poem; it shows the interconnected and overlapping nature of the structure. The second diagram shows the type of relationship that OHCO and XML require the scholar to conform to.





There are numerous ways to sidestep this problem, one of which would be to use empty elements to delineate the start and end of the word “engineers.” Empty elements are self-contained units; they are simultaneously start and an end tag. The syntax for such a tag is to insert a slash before the closing bracket: `<tag/>`. These tags do not wrap around text; often, they are used to include metadata. Using them in the above poem would look like the following:

ing bracket: `<tag/>`. These tags do not wrap around text; often, they are used to include metadata. Using them in the above poem would look like the following:

```
<line>The four<word/>eng-</line><line>ineers<word/></line>
```

When used in this way, empty elements are referred to as “milestones,” tags that serve as delineators or placeholders. While this type of solution does allow the scholar to search for the text between two empty element tags, it does not acknowledge the existence of an overlapping hierarchy of word and line elements. Essentially, it is a way of tricking the system. Since the purpose of XML is to describe a text, this does not seem like an adequate solution. Other potential solutions deal with the issue in a similar way and none reconcile the formative nature of the text with the hierarchical model of the markup procedure. For example, another popular approach, the use of fragmented markup, splits the overlapping elements into multiple parts. This approach would dictate that the word “eng-ineers” be tagged as two separate words so that there would be no overlap between the word and line elements.

Why OHCO?

There are several different reasons why digital humanists support the idea that a text can be described in terms of ordered hierarchies. The supporters of OHCO can be split into two major subgroups: those who support the model for practical reasons and those who support it for ontological reasons. Often, the line between these two groups is blurred. The ontological approach seeks to answer the question, “What is text?” and proposes the answer, “Text is an ordered hierarchy of content objects.” The logic for this argument revolves around the idea that the meaning of text can be deduced from determining the parts of a text that, if taken

away, change the text into a new text.

In a paper that seeks to defend the OHCO model, “What is Text, Really?” the authors give the following train of reasoning: “The essential parts of any document form what we call ‘content objects,’ and are of many types, such as paragraphs, quotations, emphatic phrases, and attributions. Each type of content object usually has its own appearance when a document is printed or displayed, but that appearance is superficial and transient rather than essential.”¹⁶ According to this theory, if ‘content objects’ are changed, either by removing them or changing their order, a new document is produced. By contrast, if formatting — such as font type — is changed, the text remains the same. This is similar to the classical textual editing distinction between substantive and accidental variants. Substantive variants are those that change the sense of the text, such as the substitution of words, while accidental variants are those that don’t affect the meaning of a text, such as the use of uppercase or lowercase letters. The authors go on to say that “most content objects are contained in larger content objects, such as subsections, sections, and chapters... Generally, smaller content objects do not cross the boundaries of larger ones; thus a paragraph will not begin in one chapter and end in the next. For this reason, the structure of a document is a hierarchical one, like a tree.”¹⁷

There are several problems with this reasoning. In a paper that criticizes this argument, “Refining Our Notion of What Text Really Is: The Problem of Overlapping Hierarchies,” the authors point out that even the ontological argument in favor of OHCO is motivated by practicality: “The partisans of content-oriented text processing and descriptive markup claimed that treating texts as if they were ordered hierarchies of content objects had many practical benefits, while alternative representational practices resulted in various inefficiencies and inadequacies. It was a short step from noting the practical advantages of treating texts as if they were OHCOs to explaining those advantages by the hypothesis that texts are OHCOs.”¹⁸ Interestingly, three of the authors of this paper authored, “What Is Text, Really?” three years earlier. One of the problems they point out is that the proponents of OHCO take SGML, which was not designed for fully describing texts, and attempt to justify its approach by arguing that it is practical. This is a problem because they are attempting to answer an ontological question (what is text) with

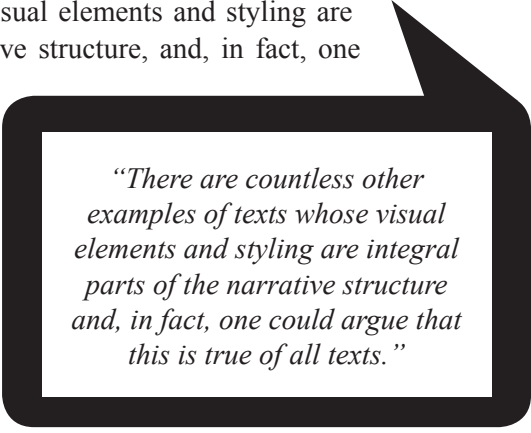
a purely pragmatic answer. “What is Text, Really?” often falls into this faulty logic. The fact that OHCO can work and that it works better than its predecessors are practical observations and have nothing to do with the nature of text. The argument that most texts can conform to a hierarchical model does not equate to “all texts are ordered hierarchies of content objects.”

There are other problems with the ontological argument that opponents do not acknowledge. “Refining Our Notion of What Text Really Is,” addresses the fact that ordered hierarchies are a debatable model, but the article does not question the idea that a text is composed of “content objects.” It makes sense that formatting would not be considered important for the original purpose of SGML, sharing law documents. However, contrary to the OHCO argument, formatting and visual presentation can be “essential” elements of a text. An example of the importance of presentation in creating meaning can be found in *House of Leaves*.¹⁹ Changing either the “content objects” or the formatting in *House of Leaves* fundamentally changes the text.

There are many visual components of *House of Leaves* that reference the narrative and are integral to the experience of reading the text. One such component is the use of different fonts. Three separate characters narrate *House of Leaves* (Johnny Truant, Zampano, and the editor), each of which is represented by a different type-font. Rather than have the narrators take control of entire chapters, Danielewski structured the novel so that each narrator often has a portion of the same chapter. If the book is reorganized so that each narrator’s story occurs in sequence, the text changes even though the component parts are still present. This is because the narrators respond to one another, and taking them out of context changes the effect of the novel. In doing so, the self-referential nature of the text is eliminated. By having different narrators comment on and add to the text, Danielewski’s novel references itself as a physical, textual artifact. In this way, the “content objects” are essential.

Unfortunately, changing the formatting would impact the novel in a similar way. Sometimes, it is impossible to identify which narrator is speaking by the content alone; eliminating the differences in type-font would make it impossible to parse certain parts of the narrative. Although the authors of “What is Text, Really?” claim that such a change would not fundamentally alter the text, one could still attempt to argue that the

formatting of *House of Leaves* fits into a hierarchical system. Such an argument would assume that the fact that the narrative contains three different type-fonts is important, but not the type-fonts themselves. This is also wrong; the type-fonts were picked intentionally and reference the character with which they are associated. For example, Johnny Truant's text is written in Courier. Within the narrative, Johnny functions as a messenger, delivering the text to the reader. There are countless other examples of texts whose visual elements and styling are integral parts of the narrative structure, and, in fact, one could argue that this is true of all texts.²⁰ While the assumption that formatting is not "essential" does not directly impact markup procedure in digital humanities, the fact that the ontological OHCO model operates on this faulty assumption indicates that the intended use of OHCO encoding is not in line with its current use.



"There are countless other examples of texts whose visual elements and styling are integral parts of the narrative structure and, in fact, one could argue that this is true of all texts."

The pragmatic argument for OHCO is not dependent on the assumption that text is composed of content objects because they are the only "essential" part of a text. Instead, this argument focuses on the practical applications of the OHCO model and argues that the success of the model means that it accurately reflects the nature of text. This approach is harder to argue with. Compared to its predecessors, SGML and XML function much more efficiently as text encoding languages.²¹ Revisions of the OHCO thesis also suggest a way in which overlapping hierarchies are not at odds with the OHCO model, a tempting argument given its practical advantages.

In this argument, hierarchies that overlap are considered to belong to a different sub-hierarchy: *x* is a sub-perspective of *y* if and only if *x* is a perspective and *y* is a perspective and the rules, theories, methods, and practices of *x* are all included in the rules, theories, methods, and practices of *y*, but not vice versa.²² By this logic, any overlapping hierarchy can be decomposed into sub-perspectives of the text. However, the

“rules, theories, methods, and practices” mentioned seem completely open to interpretation. By this definition, it seems as if any perspective could be ruled a sub-perspective of another perspective.

The practical benefits of OHCO combined with this revision might initially seem to eradicate doubts about the validity of the model. Upon a close examination, however, problems still exist. The authors of “Refining our Notion of What Text Is” point out that there are cases in which overlapping hierarchies are created by objects that self-overlap, or overlap with more of the same type of object. For example, if a question overlaps with a line in a poem, it could be argued that questions and lines are elements of different perspectives. If, on the other hand, two stories overlap with one another, this argument falls apart.

Even without contemplating the existence of self-overlapping objects, the revision is hard to digest. The creation of sub-perspectives seems ambiguous and arbitrary. Even if it is true that overlapping hierarchies are really perspectives and sub-perspectives, the coding solutions remain the same. Either perspectives and sub-perspectives can be circumnavigated with coding tricks, or, perhaps more in line with this revision, different perspectives can be marked up in multiple copies of a document. These solutions are not adequate; if there is still no way to represent “perspective and sub-perspectives” as coexisting features of the same document, the revision is useless.

Despite these difficulties, ultimately, the practical benefits of OHCO are real. While the ontological argument has more obvious flaws, the practical approach is grounded in almost ten years of largely successful XML encoding in the humanities. However, there is a difference between acknowledging the usefulness of these languages and assuming that their superiority over previous models precludes the development of new and better technologies not based on the OHCO model. In light of the difficulties posed by overlapping hierarchies, it seems counterproductive that both the ontological and pragmatic arguments for OHCO cling to the validity of the current markup model.

The Appeal of OHCO

If there are so many issues with the OHCO argument, where does the inclination to treat it as a definitive and infallible model come from? A

good insight into this question might come from the way that scholars feel about the process of text encoding. Though “feel” is hard to measure, Stephen Ramsay, a programmer and digital humanist, offers some particularly useful insights. Ramsay refers to the encoding process as an act of creation: “As humanists, we are inclined to read maps (to pick one example) as texts...But *making* a map (with a GIS system, say) is an entirely different experience...Building is, for us, a new kind of hermeneutic — one that is quite a bit more radical than taking the traditional methods of humanistic inquiry and applying them to digital objects.”²³

Comparing encoding to building is particularly useful in that it highlights the relationship between the scholar and markup language as that of builder and tool. As with most tools, if they get the job done, there is no reason to question them. XML fits into this category, as it allows for the categorization of textual features and the production of digital texts. In addition, the basic syntax of XML makes the language seem deceptively open ended. Because the scholar has complete control in determining tag names and in determining which textual elements to tag, the limitations of the language are not always obvious. Also, current approaches for dealing with overlapping hierarchies are very easy to implement, even though they do not fully address the problem.

Another helpful way of thinking about the allure of OHCO might be as follows. For a long time, people thought CD players were wonderful. Although CD players could not play videos, this was not recognized as a “problem” or inadequacy, because a video playing CD player was unimaginable. Most owners of portable CD players did not stop to question the fundamental design properties of the tool they were using to listen to music. The people who did stop to consider the fundamental design principles of CD players were the engineers making them: people who were intimately familiar with the basic principles behind the creation process. Eventually, the iPod came out and changed the way people viewed their music players. Engineers were able to conceive of a future technological advance because they understood the basic design principles behind it and could determine what was feasible.

Most digital humanists do not fully understand the tools they are using or the principles behind them, nor are they aware of the history of SGML and XML. They know that, for the most part, the tools work. Humanities scholars, the people who care about the ontological aspects

of the markup model, cannot understand how another model might exist without better understanding the current system. Thus, many scholars fall into the trap of unquestionably accepting the OHCO model. Scholars need to be aware of the fact that the tools they are using were not designed for them, and that other options are available. This does not mean that scholars should disavow the current system, but, in order to improve, there needs to be more room for inquiry.

Alternatives and Conclusions

CONCUR, an alternative to the OHCO approach, was originally a feature of SGML but has fallen out of use, mainly because it creates cumbersome and verbose markup. This drawback should not automatically invalidate its conceptual value; it is based on the principle of concurrent data structures, a way of dealing with multiple threads of data at the same time. CONCUR allows for a document to be simultaneously marked up in multiple conflicting hierarchical tag-sets. It was not included as a feature in XML, and while some similar languages have been developed, none are widely used in humanities markup.²⁴

Although there are a number of problems with CONCUR, its basic principle — acknowledging and supporting multiple hierarchies simultaneously within a document — could be a useful way of dealing with the inadequacies of the current model. In addition, the principles behind CONCUR may more adequately reflect the relationship between interpretations of a text. It follows that perhaps such a model might also allow for a better relationship between scholars and the markup procedure they are using to describe a document.

However, with that said, it is important not to adopt the same route as the proponents of the OHCO model. The fact that CONCUR might more conveniently or completely represent certain features of a text does not mean it represents the true nature of what text “really is.” CONCUR, like standard SGML and XML, assumes that a text is composed of hierarchies. While this assumption is not necessarily wrong, emerging markup theories should not assume any principles of previous encoding methods as a given and should be discussed critically from the perspective of the humanities scholar.

The “problems” with the current model of textual markup, em-

FORBES & FIFTH

bodied by the issue of overlapping hierarchies, indicate a more fundamental issue with the way text encoding has, for the most part, been conceptualized. This is a serious problem because the constraints of markup strategies have serious, real world ramifications in the relationship among text, scholar, and reader. If scholars must fit a text they wish to encode into a hierarchical structure, does this mean that scholars are ignoring significant textual relationships that do not conform to the parent, child, and sibling model? Because they were not originally intended for literary scholarship, SGML and XML cannot be assumed to represent the “true” nature of text, nor can they be assumed to be the best languages for describing texts. The fact that these markup languages have been used successfully does not indicate that there is not room for significant improvement. In light of the history and limitations of current markup procedure, literary scholars need to reclaim their methods of transcription. This might not mean that the OHCO model for text encoding needs to be abandoned. However, it is impossible to determine the correct approach to text encoding without opening up a larger discussion about what digital humanists want and need their markup languages to provide.

