DHQ: Digital Humanities Quarterly
2019
Volume 13 Number 3

A Model of Versions and Layers

Desmond Schmidt <desmond_dot_allan_dot_schmidt_at_gmail_dot_com>, Charles Harpur Critical Archive

Abstract

Our libraries are full of manuscripts, many of them modern. However, the digitisation of these unique documents is currently very expensive. How can we reduce the cost of encoding them in a way that will facilitate their study, annotation, searching, sharing, editing, comparison and reading over the Web? Unlike new documents prepared for the Web, historical manuscripts frequently contain internal variation in the form of erasures, insertions, substitutions and transpositions. Variation is also often expressed externally between copies of one work: in successive print editions, in manuscript copies or successive drafts. Current practice is to prepare separate transcriptions of each physical document and to embed internal variation directly into the transcribed text using complex markup codes. This makes the transcriptions expensive to produce and hard to edit, limits text reuse and requires that transcriptions be first disentangled via customised software for reading, searching, or comparison.

An alternative approach, described here, is to separate out the internal variation of each document into notional layers. This is done primarily in order to facilitate the recording of revisions at any one point. The move is, of course, counter-intuitive since these document-wide layers were not intended by the author as texts to be read. But it proves itself in practice by radically simplifying the tasks of editing, searching and comparison. Versions, on the other hand, are higher-level constructs that do represent the state of a text as it was left at some point in time by an author or scribe.

By employing layers to record complex revisions, the task of computing differences among intra-document layers and against versions of the same work in multiple documents may be delegated to the machine rather than having to be recorded laboriously by hand. The ensuing simplification of markup reduces transcription and editing costs, boosts text reuse and searching, and, by removing the need for customised software, increases the longevity of digital transcriptions.

1. Introduction

The task of the modern editor often starts with a digital recording of the contents of original documents written on a variety of physical media. If produced by a machine, such as a printing press, text presents itself for encoding as a sequence of characters arranged into evenly spaced lines contained in one or more columns per page. In practice, though, even the printed text of a single document will contain many errors: obvious misspellings, gaps, duplicated words, misprints etc. [Shillingsburg 1996, 134f]. Depending on its age it may also contain ligatures and abbreviations whose expansion into more modern forms may seem necessary to its editor [Greetham 1994, 159]. The corrected and expanded text thus represents another version of what seemed at first glance to be easily representable via a simple sequence of character codes. 1

Another class of documents that contain hidden internal variation is the professionally produced medieval manuscript. Trained scribes worked in scriptoria to laboriously copy out each line neatly for reading. Although they were careful, mistakes were common. Haplography or dittography are two well-known types (where material was inadvertently left out or added). Incorporating readings from a second exemplar when the first one was deficient, corrections by a second hand and of course numerous ligatures are just some of the textual phenomena that complicate the transcription process [Greetham 1994, 279ff]. 2

A further type of document containing internal variants is the modern holograph manuscript revised by its author. Since in most cases the changes are in the same hand and ink-colour, it may be difficult to discern a clear sequence of revisions. The writer may have tinkered with the text over an extended period of time. The result may not be entirely legible, but even if it is, unravelling the sequence of script-acts that created it is usually conjectural [Pierazzo 2009]. 3

To transcribe each text fully, no matter its provenance, thus requires some way to represent textual variation both within the document and also between documents: original and successive manuscript drafts, fair copies and, for printed works, typescripts and proofs, first and later editions. 4

There is still no satisfactory commercial solution to these two linked problems, and even if there were, any tools would necessarily be subject to the commercial software lifecycle, which for major products is estimated to last around 8–12 years [Sandborn 2012, 143–155]. Although open source software is immune to some of its effects, it is still subject to platform and media obsolescence, that is, to the obsolescence of dependent services and tools [Jenab 2014]. This situation is not at all what humanists want. Ideally, they would like digital texts and the software that goes with them to have the same stability as printed books. 5

Our libraries are replete with manuscripts. For example, the National Library of Australia has 16 kilometres of shelf space dedicated to manuscripts [NLA 2016], and other large libraries have similar holdings. What is to become of even the smallest proportion of this material deemed worthy to be digitised one day, not as hard-to-read image files, but with their contents transcribed for editorial purposes and sophisticated end-use? Will they have to be remade every 10–20 years because technology has moved on? In that case the cost alone of maintaining existing transcriptions and software is likely to consume all available resources. By contrast, the shelf life of the printed book extends back to its inception in the 15th century. Anyone who has handled an incunable will marvel at the quality of the paper, the clarity (if antiquated appearance) of its type. How can our digital texts even begin to approach this kind of longevity? 6

The problem reduces to this: any technology beyond the most common and stable variety is doomed to become obsolete within its normal lifecycle of around 10–20 years. Beyond the mere recording of data, customised computer programs required to manipulate and present the textual data will become obsolete even sooner without constant maintenance. Continuously seeking after a supposedly final technology that will immortalise new editorial work is thus unlikely to succeed. In practice, change is constant, and may come so fast that the chosen technology becomes obsolete even before an editorial project ends [Göbel 2015]. 7

Is there any answer to this problem, or is it simply the natural property of digital text and the software needed to manipulate it that renders both ephemeral? Fortunately, certain digital technologies seem to endure far longer than others. For example, the Unicode standard arose by incorporating earlier smaller character encoding standards like ASCII (1963) then ISO-8859-1 (1987), culminating in the Unicode standard itself (1991). To change the assignation of letters to codes in this basic character set would invalidate billions of texts worldwide for no overall benefit. So it is not far-fetched to suppose that in, say, 100 years time, a digital file containing nothing but Unicode character codes will continue to be readable. 8

Accordingly, what is proposed here is a simple use of the most basic and stable digital technologies to encode the contents of complex historical documents. If the approach is successful, any software of current or future design would then be able to manipulate those documents into useful forms. The guiding idea is that the less technology we put into our digitally recorded texts the more useful and durable they will be in the future. 9

This proposal has nothing to do either with the technology used to manipulate encoded texts, or with the editorial method for creating them. It is orthogonal to both; and is, rather, a simple but effective strategy for recording transcriptions of complex original documents in digital form. To explain how this works in practice, the second section below develops a model of versions and layers through a series of real-world examples. The third and fourth sections then describe the advantages and disadvantages of this rearrangement of digital textual data as it relates to the scholarly edition. Finally, the fifth section explains in abstract terms how these texts can be used to make digital scholarly editions. 10

2. A model of versions and layers

This model of versions and layers originally developed out of the author's own work on the manuscripts of Ludwig Wittgenstein [Nedo 1993, 73–77]. It is most closely related to what Hans Walter Gabler calls the "German school of editorial scholarship", including the historical-critical edition, in which the "critical" component referred not so much to the establishment of the text but to the analysis or critique of the text's genesis and history [Gabler]. Gabler perhaps coined the term "layers" in his 1986 edition of Ulysses [Gabler 1986] although the concept seems to apply only to levels of local revision. The same concept was taken over into the HyperNietzsche Markup Language [Saller 2003]. A similar approach is also taken in genetic editions like Dietrich Sattler's Hölderlin, where the apparatus is placed inline by graphically arranging variants above the line of text they pertain to. As Sattler says, this results in a text where the variants are readable in their true context, not removed to a distant apparatus [Sattler 1979]. The concept is basically the same here, except that it has been translated from the printed page into a simple arrangement of plain-text files. 11

Out of the three classes of texts described in the introduction: printed books, medieval manuscript copies of earlier texts and modern holographs, the examples explored below use mostly the third category. This is because the model pertains to the process of writing itself, not to a particular text-type. Holographs were chosen because they typically exhibit higher levels of internal variation than the other two, and for the model to be universal it has to cater for the worst-case scenario. This is not meant to imply that the three classes of text are equivalent, except in so far as they all represent the phenomena of writing or printing on paper, parchment or other physical surfaces. 12

The examples are mostly taken from the manuscripts of the Australian poet Charles Harpur (1813-1868). The Charles Harpur Critical Archive [Eggert 2019] has obtained permission from their owner, the State Library of New South Wales, to publish the images and content of all Harpur-related material in its possession. Also, thanks to recent changes in Australian copyright law, from 2019 unpublished manuscripts will have the same copyright duration — 70 years after the death of the author — as printed works [NLA 2017]. So these texts form a very useful set of freely available examples from which to develop the model described here. They are also very varied in nature, consisting of newspaper cuttings, printed pamphlets and books, drafts of corrected manuscripts and letters. As will be demonstrated below, only a very small number of basic editing operations on handwritten texts are needed to put the proposed model of versions and layers into practice. The focus on Harpur does not introduce a bias because all the textual phenomena being explored here [1] are common to virtually all handwritten texts, and examples of each phenomenon have been verified as also existing in the manuscripts of Melville's *Typee* [Bryant 2009], Wittgenstein [Pichler 2016], Samuel Beckett [Van Hulle 2015], the Faust Edition [Bohnenkamp 2016] and the Shelley-Godwin Archive [Fraistat 2018]. The markup schema described in the Text Encoding Initiative Guidelines Chapter 11 "Transcription of Primary Sources" provides higher-level or finer grained descriptions of these same basic textual phenomena [TEI 2017]. They can also be found in Bryant *The Fluid Text* [Bryant 2002]. 13

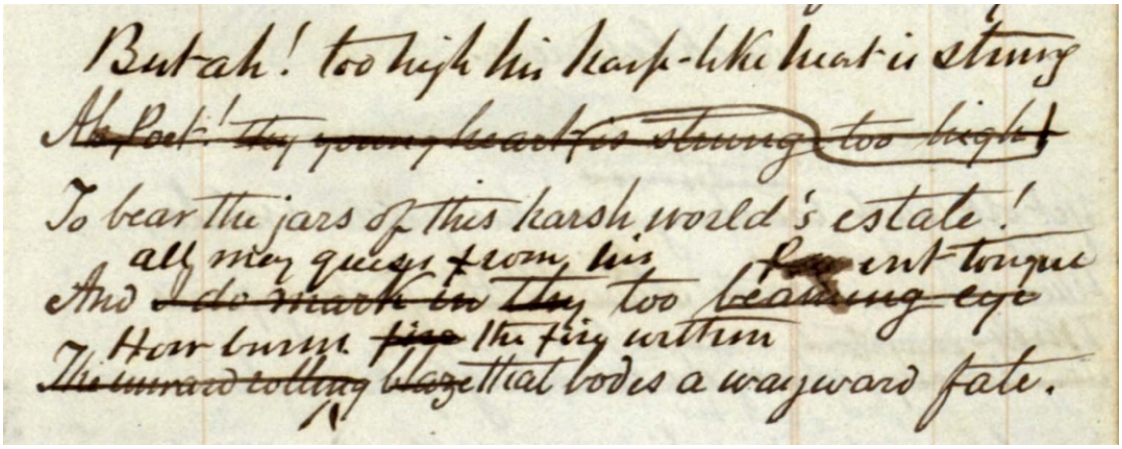

Figure 1.

Figure 1 shows one stanza taken from Harpur's poem "Vestibulary Stanzas" in manuscript B78 written in 1855 [Harpur 14

1855, 11]. Seven changes can be discerned:

line 2: the transposition of “is strung” and “too high”

line 2: the replacement of the entire line by line 1

line 4: the replacement of “thy” by “his”

line 4: the replacement of “I do mark in” by “all may guess from”

line 4: the replacement of “beaming eye” by “fervent tongue”

line 5: the replacement of “blaze” by “fire”

line 5: the replacement of “The inward rolling fire” by “How burns the fire within”

The temporal order of these changes can be determined in only two cases. It is clear that the transposition in line 2 preceded its replacement by line 1. Also the replacement of “blaze” by “fire” in line 5 preceded the replacement of “The inward rolling fire” by “How burns the fire within”. Other than these inferences, the exact sequence of changes here is unrecoverable. For instance, it is not clear if the poet changed line 2 before lines 4 and 5. All remaining changes have thus to be treated as independent of one another. 22

Some definitions may now be put forward: 23

Definition 1: A change is *independent* if it can be carried out without affecting the sense, grammar or metre of the rest of the text. 24

Definition 2: A *state* is a hypothetical or actual reading of part of the text before or after one or more independent changes have been carried out or not. [2] 25

For example, line 4 may have once been in any of the following states: 26

And I do mark in thy too beaming eye
And I do mark in thy too fervent tongue
And I do mark in his too beaming eye
And I do mark in his too fervent tongue
And all may guess from thy too beaming eye
And all may guess from thy too fervent tongue
And all may guess from his too beaming eye
And all may guess from his too fervent tongue

All possibilities are equally valid here, but the final “state” of the entire line is clearly “And all may guess from his too fervent tongue”. 27

Also, following Bryant’s terminology, we offer: 28

Definition 3: A *revision site* is a *hot-spot* [Bryant 2002, 151] or cluster of dependent changes at some point in the text-stream. 29

In Figure 2 the substitutions and deletions “the/his”, “whereon they/on which his spirit”, “to burn”, “the/some”, “to burn” and “,” are revision-sites. Changes that are local to one revision-site are less likely to be connected with changes elsewhere in the same manuscript the further apart they are. A change at the start of a document is very probably (although not certainly) unrelated to a change at its end. Even changes in the same stanza, as in the example above, are more likely to be unrelated than related. 30

Figure 2, from the same manuscript [Harpur 1855, 75], is an example where several changes are related by grammar and sense. 31

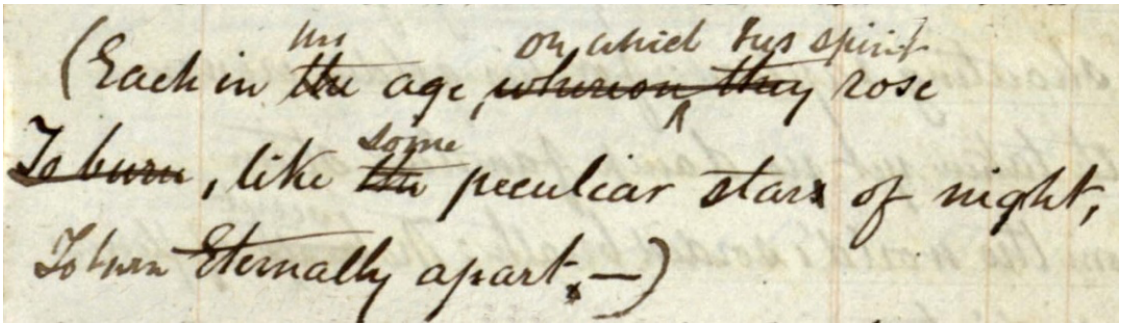

Figure 2.

The two layers here are: 32

1:
(Each in the age, whereon they rose
To burn, like the peculiar stars of night,
Eternally apart, —)

2:
(Each in his age, on which his spirit rose
like some peculiar star of night,
To burn Eternally apart —)

The substitutions: “whereon they” by “on which his spirit” and “stars” by “star” are related by grammar, and were presumably carried out at the same time, not independently — the plural “stars” according with “they” and not “his”. Similarly, the deletion of “To burn” and its reinsertion on the final line are related by sense. However, each of these two sets of changes taken as a whole is independent of the other, and thus acts as a single independent change. 33

This independency of individual or grouped dependent changes has an unfortunate mathematical consequence for storage. Each independent change consists of at least two states (carried out or not), so over the entire document the number of possible texts obtainable will be at least 2×2...or 2^N, where N is the total number of independent changes. In the single stanza in Figure 1, as described above, there were 3 independent changes and two pairs of temporally dependent changes, each having 3 states, for a total of $2^3 \times 3^2 = 72$ possible texts. Over an entire document the total number of possible texts will be exponentially large. For just 100 independent changes we have 2^{100} or approximately 1.267×10^{30} possible texts or 1267 with 27 zeroes after it. Many documents will contain far more changes than that. So any attempt to store all possible states of a document’s contents in this exhaustive fashion is bound to fail. 34

2.1 Layers

As already noted, the changes in lines 2 and 5 in Figure 1 are related in time. In certain texts there may be temporal sequences of up to 10 or more local substitutions, as in the Magrelli poem “La campagna Romana” [Fiormonte 2010]. Each change belongs to a local point in time or level, and the state of the text at each stage includes the context that surrounds it. Since it is not possible to record all local states separately in their version-wide contexts (because they are exponentially too many) a simple solution is to assign all changes in the document occurring at the same position in the *local* temporal sequence to the *same* numbered layer. 35

Definition 4: A *layer* is a tracing of the documentary text from start to finish that includes all the changes in each revision-site at a given level, the unchanged text that lies between them and the final state of other revision-sites that end at a lower level. 36

So instead of 72 layers for this stanza — the total number of possible combinations of independent changes — only 37

three are needed: the initial state of each segment of text that undergoes change and its subsequent states. For example, line 2 admits of three such states:

1: Ah Poet! thy young heart is strung too high

2: Ah Poet! thy young heart too high is strung

3: But ah! too high his harp-like heart is strung

Thus, to record the stanza's text in its entirety requires only three layers. The unrelated changes noted above require only 2 layers each, and line 5 requires 3. If all the local states at a particular level are merged into one layer, then each layer becomes a *record* of the local states of change. Layer-1 would be the text as it was originally throughout. Layer-2 is its state after the first local change, and Layer-3 or Layer-final its state after a second local change. This makes no claim that Layer-1 or Layer-2 ever existed as a version-wide text: more on this, below. It is merely a way of recording and storing all of the text of this complex document using only the technology needed to record unchanged text. For clarity, here are all the layers of Figure 1 written out in full: 38

1. Ah Poet! thy young heart is strung too high
   To bear the jars of this harsh world's estate!
   And I do mark in thy too beaming eye
   The inward rolling blaze that bodes a wayward fate.

2. Ah Poet! thy young heart too high is strung
   To bear the jars of this harsh world's estate!
   And all may guess from his too fervent tongue
   The inward rolling fire that bodes a wayward fate.

3. But ah! too high his harp-like heart is strung
   To bear the jars of this harsh world's estate!
   And all may guess from his too fervent tongue
   How burns the fire within that bodes a wayward fate.

An obvious objection here is that the author almost certainly never wrote layers 1 and 2. Mathematically, the chances that he did so in this case are just 1 in 72. However, so long as it is remembered that the only layer that ought to be considered a "version" is "Layer-final", the objection is moot. Layers are merely a mechanism to record local changes, *not* a record of how the scribe or author wrote the document as a whole. They do not constitute an ontological claim. So long as it is remembered which transcriptions are layers, and which can be considered versions, no confusion should arise. 39

2.2 Deletion, insertion, substitution and transposition

The key advantage — the true efficiency — of layers is that they do away with the need to explicitly record deletions, insertions, substitutions and transpositions. Any text present in one layer and absent from a subsequent layer can be assumed to be deleted. Likewise any text that appears in a subsequent layer, where it was previously absent, is an insertion. Text that is replaced by other text at the same location in a subsequent layer can be regarded as substituted. Similarly, text that appears transposed in a later layer can be considered transposed. On the other hand, using current encoding techniques, text must be manually labelled as *deleted*, *inserted*, *substituted* or *transposed* using explicitly entered, complex markup codes [Pierazzo 2014a, 12]. This considerable human effort is not needed if it can instead be undertaken by a computer. 40

Furthermore, by marking the text as *deleted* or *inserted* etc., current digital encoding practice conflates the membership of an internal variant in a particular layer with its graphical status, in effect declaring that deleted text is in a *crossed-out* format, just as, say, underlining is a format. But, unlike underlining, crossed-out text is also replaced, perhaps by nothing (a deletion) or by something else (a substitution). That these two operations — the assignment of a format and the substitution — are separate and distinguishable is proven by the case of open variants, where the earlier version of the 41

text is not crossed-out [Van Hulle 2011, 804].

An example in Harpur can be found in “The Bard of Paradise” [Harpur 1863, 4]: 42

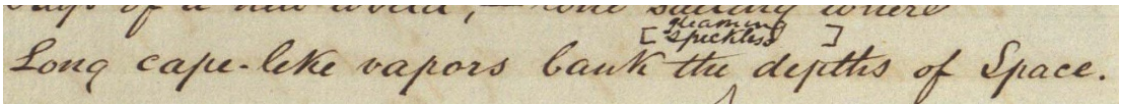

Figure 3.

Harpur first wrote “speckless” before squeezing in “gleaming” above it. But “speckless” is not crossed-out and the added pair of brackets indicates that the author considered them both viable alternatives that, in this manuscript, he did not go on to resolve. For the proposed layering approach the usual assumption that text in an earlier layer is deleted may be resolved via markup. By this means the earlier variant can be labelled as “undeleted” without overly complicating the encoding of the layer itself. As we have seen, the layering approach is justified by its efficiency: even in the very uncommon case of open variants when markup cannot be avoided considerable savings in coding can nevertheless be realised. 43

2.3 Versions

Although the term “version” has been used variously by scholarly editors, the definition proposed here is practical and simple: 44

Definition 5: A *version* is a single document-wide state of the text, determinable with reasonable certainty, in which the author or scribe completed or abandoned the entire documentary text at some point in time. 45

Although in most holographs the only determinable version is the final layer or state of a manuscript, in some cases an individual manuscript may be witness to more than one version. This section describes some examples of this. 46

Harpur’s first draft of his sonnet “Trust in God” was written on the inside back cover of a copy of his 1853 book *The Bushrangers* [Harpur 1853a]. He then redrafted it on the inside front cover. Figure 4 shows this latter text, which is mostly written in a dark-coloured ink with some corrections in lighter ink. Disregarding the lighter corrections, the final layer of the dark-ink version corresponds closely to the copy published in the *Empire* newspaper of 20 June 1853 as shown in Figure 5 [Harpur 1853b]. A subsequent publication in the *Australian Home Companion* of 1859 (Figure 6) accords with the state of the text after the final corrections in lighter ink, with some changes to the pentultimate line [Harpur 1859]. It can thus be seen that this manuscript contains two discernable document-wide states of the text – two versions – each of which has its own local corrections or layers. 47

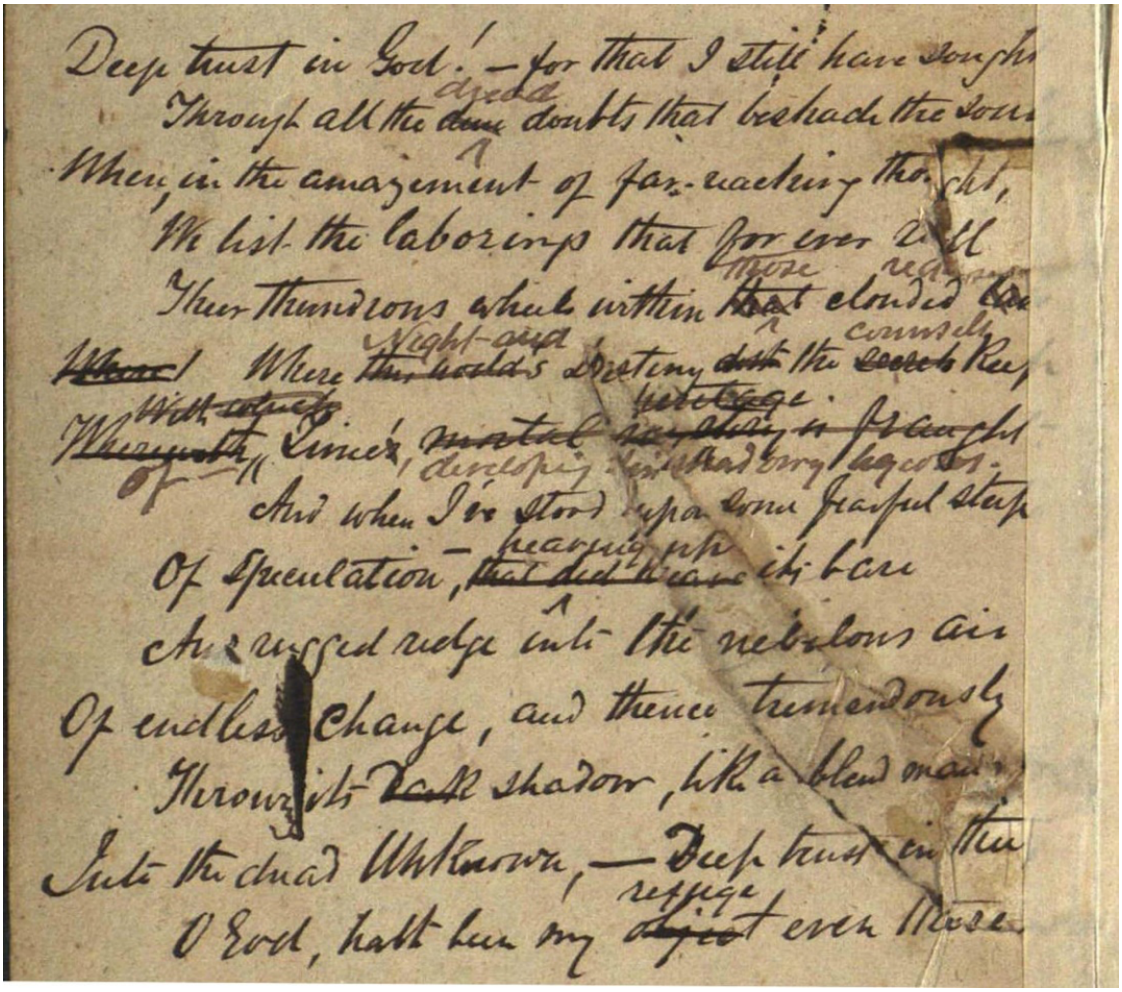
Deep trust in God! — for that I still have sought
Through all the dim doubts that beshade the soul
When, in the amazement of far-reaching thought,
We list the labourings that forever roll
Their thundrous wheels within that clouded lair,
Where this world's Destiny doth the secrets keep
With which Time's mortal heritage is fraught.
And when I've stood upon some fearful steep
Of Speculation, that did heave its bare
And rugged ridge into the nebulous air
Of endless Change, and thence tremendously
Throw its dark shadow, like a blind man's stare,
Into the dread Unknown, — Deep trust in Thee
O God, hath been my refuge even there.

Figure 4.

Deep trust in God!—for that I still have sought
Through all the dim doubts that beshade the soul,
When, in the amazement of far-reaching thought,
We list the labourings that forever roll
Their thundrous wheels within that clouded lair,
Where this world's Destiny doth the secrets keep
With which Time's mortal heritage is fraught.
And when I've stood upon some fearful steep
Of Speculation, that did heave its bare
And rugged ridge into the nebulous air
Of endless Change, and thence tremendously
Throw its dark shadow, like a blind man's stare,
Into the dread Unknown!—deep trust in Thee,
O God, hath been my refuge even there.

Figure 5.

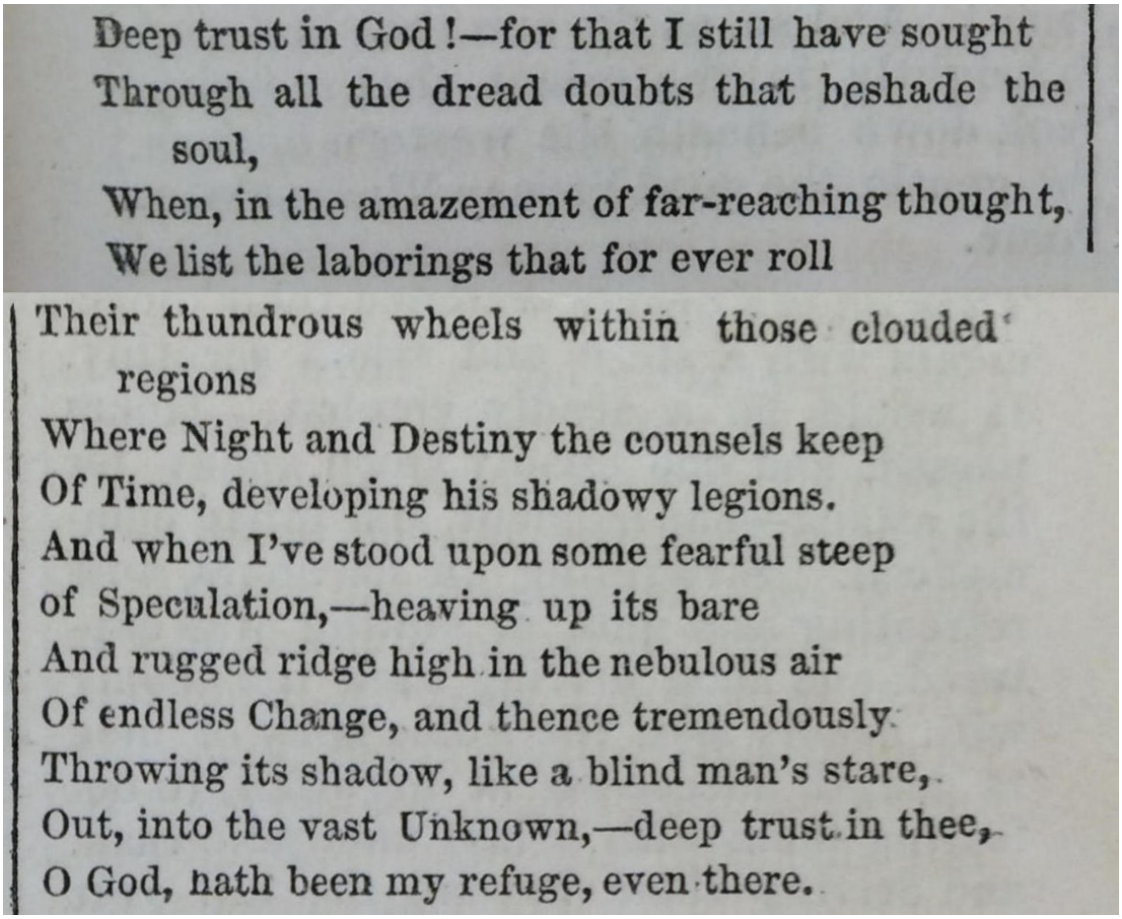
Deep trust in God!—for that I still have sought
Through all the dread doubts that beshade the soul,
When, in the amazement of far-reaching thought,
We list the laborings that for ever roll
Their thundrous wheels within those clouded regions
Where Night and Destiny the counsels keep
Of Time, developing his shadowy legions.
And when I've stood upon some fearful steep
of Speculation,—heaving up its bare
And rugged ridge high in the nebulous air
Of endless Change, and thence tremendously
Throwing its shadow, like a blind man's stare,
Out, into the vast Unknown,—deep trust in thee,
O God, hath been my refuge, even there.

Figure 6.

Professionally produced manuscripts may contain similar internal versions. Many medieval manuscripts were corrected at different times, sometimes by a corrector by comparison with an exemplar or with another manuscript [Pauly 2003, Copy]. An example known to the author is the Ancient Greek Bacchylides papyrus written in the first century BC [Jebb 1905, 126], a section of which is shown in Figure 7 [Jebb 1905, plate 1].

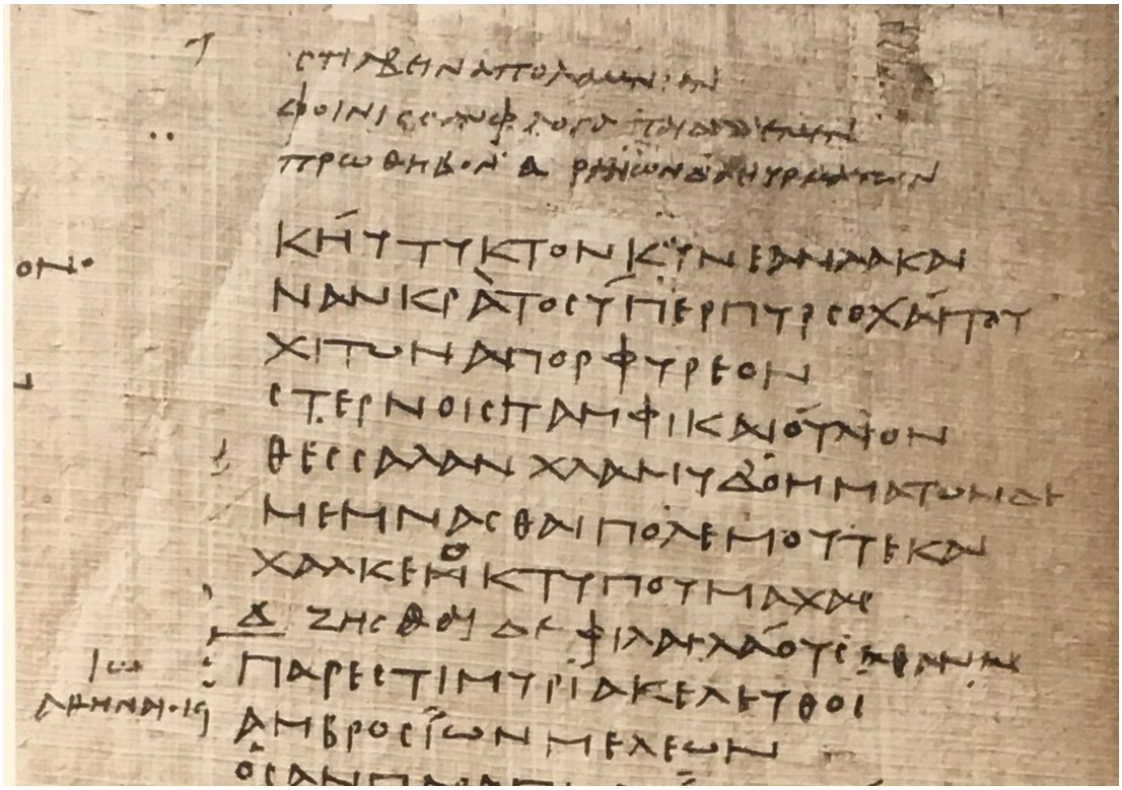

Figure 7.

The baseline text A was revised in three distinct hands A^1, A^2 and A^3. A^1 is probably the scribe himself, fixing simple mistakes of spelling. A^2 is probably a different but contemporary hand in which titles are added in the margins. A^3 is a Roman cursive hand of the second century or later [Jebb 1905, 133]. So A^1 is probably a layer of the first version, while A^2 and A^3 form two new versions, which subsume the final corrected texts of A^1 and A^2 respectively. In Figure 7 the three lines inserted at the top are by A^3 (a faint marginal mark inserts them before line 6), the title in the left margin *ΙΩ ΑΘΗΝΑΙΟΙΣ* was added by A^2 and the correction of *ΧΑΛΚΕΝΤΥΠΟΥ* to *ΧΑΛΚΕΟΤΥΠΟΥ* in line 7 was by A^1. 49

Bryant likewise sees three internal versions in Melville's *Typee* manuscript, which he calls "transcription, transformation, and translation" [Bryant 2008]. Van Hulle and Neyt call internal versions "revision campaigns" when they can be reliably distinguished by a new writing instrument or change of ink [Van Hulle 2015]. 50

However, in the absence of such information about discernible pens or differences in handwriting or style most holographs contain only one version. It is up to the human editor to decide whether a correction belongs to a specific layer or version. 51

3. Advantages of versions and layers

This model of versions and layers has certain advantages. These can be appreciated through a comparison of the versions and layers model with the current method of recording variants in manuscripts using inline embedded markup. 52

3.1 Problems with embedded markup

In the embedded markup method variants within a document are inserted directly into the text-stream wherever the transcriber encounters a revision-site. 53

Many markup languages have been proposed that adopt an embedded or inline approach to recording variants. (In what follows the terms "inline" and "embedded" are used interchangeably.) One of the earliest was MECS (*multi-element encoding system*), which evolved from the Wittgenstein markup developed at Tübingen in the 1970s [Huitfeldt 54

1993]. The same technique was later used in the TEI (Text Encoding Initiative) Guidelines, and in many ad hoc markup systems developed for individual editions, such as the Tagore Bichitra [Tagore]. In order to illustrate the disadvantages of inline recording of variants, a simple hypothetical markup language may be used that maintains the generality of the argument, while ensuring clarity, without reference to any specific system of markup. The language is defined by four rules:

1. Deletions are enclosed in "[" and "]".
2. Insertions are enclosed in "<" and ">".
3. Insertions and deletions may nest: inner ones are earlier than those on the outside.
4. Insertions and deletions that immediately follow one another with no intervening spaces indicate a temporal succession from left to right.

Although they may use different syntaxes these simple rules are followed by all markup languages that are used to record inline variants. For example, the text of Figure 1 could be transcribed in this markup language as follows: 55

> [Ah, Poet! thy young [heart is strung too high]<too high is strung>]<But ah! too high his harp-like heart is strung>
> To bear the jars of this harsh world's estate!
> And [I do mark in]<all may guess from> [thy]<his> too [beaming eye]<fervent tongue>
> [[The inward rolling][blaze]<fire>]<How burns the fire within> that bodes a wayward fate.

The presence of the embedded variants here makes it difficult for a human editor to see in what state the text was at any stage in its revision, and requires him or her to visualise a complex series of steps using abstract symbols. The heavier the corrections the harder this task becomes. Occasionally our trained Harpur transcribers would even decline to encode Harpur's more complex passages. For instance, one noted: "Lines 15 and 31 contain numerous additions and deletions; only the final intentions have been transcribed here." 56

Searching will also be impaired, since the inline variants interfere with the retrieval of literal expressions. Search engines typically don't see the markup. All they see is the text in the nonsensical word-order given above. As a result, searching for the literal phrase "All may guess from his too fervent tongue", as finally formulated by the author, will *not* be found, while the expression 'rolling blaze fire How burns the fire within' *will* be found, even though the author never wrote it. [3] 57

When comparing two versions of a text, inline variants get in the way and must first be removed [Juxta 2013][Dekker 2015]. Computer algorithms commonly used in the humanities for collating texts such as Heckel (1978) and Myers (1987) are not designed to handle inline variants. And no one has yet demonstrated a reliable method for comparing texts that contain them, or proven that it is technically feasible. [4] 58

3.2 Difficulty of overcoming these problems

One obvious solution to some of these problems would be to write a program to extract coherent texts from the embedded encoding, retrieving either the first, last or intermediary stages in the text's evolution. However, in practice this is either difficult or impossible to achieve, for three main reasons. 59

3.2.1 *Currente calamo* corrections

The original state of a text before any changes were made might seem to be one that could be readily recovered from an embedded encoding of variants, by simply accepting only first-level variants and simple deletions, and rejecting all others. However, the presence of corrections made in the process of writing, or *currente calamo*, as shown in Figure 8 [Harpur 1863, 359], makes this difficult or impossible. 60

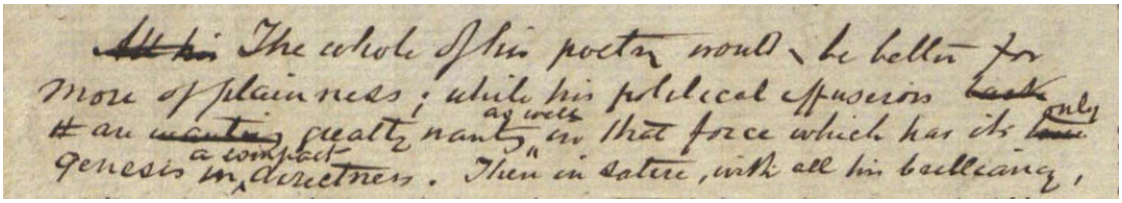

Figure 8.

Here the author progressively revised the text as he wrote it. He started the sentence with “All his”, then corrected this immediately to “The whole of his poetry”. Later he started writing “his political effusions lack th...” then corrected this to “his political effusions are wanting...” then corrected the last two words to: “are greatly wanting”. All these changes were carried out *on the baseline*. Simply encoding them as deletions — as they appear on the page — and then writing a program to extract the “original” text before deletion would result in the nonsense: “All his The whole of his poetry would be better for more plainness; while his political effusions lack th are wanting greatly wanting....” 61

3.2.2 Interpretation in markup

Marking up a text is generally agreed to be an interpretative act [Sperberg-McQueen 2000] [Durusau 2006] [Bauman 2011]. Interpretation must therefore govern the choice of codes and their application to the representation of inline variants. Extracting coherent texts for searching, reading, or comparison from such transcriptions thus requires, at the very least, the writing of customised software for each set of documents encoded in a project-specific set of conventions. This is further complicated by the fact that even trained encoders have different understandings of what they are looking at on the page and therefore how to encode it. They may even change their encoding practice from day to day [Durusau 2006]. For Harpur we found that over 30 different ways had been used by just three transcribers to record second-level corrections. Even if an encoding can be made syntactically consistent we must ask can it also be made *semantically* consistent: i.e. does each specific instance of an encoding always mean exactly the same thing, and is the same kind of textual phenomenon always encoded in the same way? 62

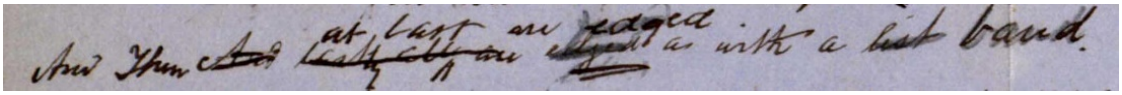

Figure 9.

An example of the potential for semantic variation is shown in Figure 9. Taken from an early manuscript of “Dawn and Sunrise in the Snowy Mountains” [Harpur 1849-63, 573], the revisions at the start of the line might be validly encoded in our simplified markup language as “And Then [And]” or equally as “<And Then>[And]” or as “<And <Then>>[And]”. However, the use of capitals at the start of all three words indicates that Harpur revised the text by gradually adding words to the left, so a better encoding might be to reverse the order: [And][<Then>]<And Then>. The question then becomes, how can software process alternative formulations of the same thing in a consistent fashion, and how can it determine the context of each change? 63

3.2.3. Conflict between graphical and temporal encoding

Recording the graphical appearance of the text on the page via the embedded markup method described above makes the extraction of the temporal succession of local changes virtually impossible [Schmidt 2014, §5]. For example, most transcribers would record line 1 of Figure 1 before line 2, but the temporal sequence is actually 2, 1. Another example is the use of inserted blocks of text. Like many authors, Harpur sometimes redrafts heavily revised portions of his text in the margins or on separate pages, as shown in Figure 10 [Harpur 1855-62, 44–45]. Due to the complex encoding required for inline variants in such cases, encoders tend to record such blocks separately and to specify their graphical layout (e.g. position, orientation) rather than integrate the textual changes they contain as part of the main text [Muñoz 2015], and then record their temporal sequence, if ascertainable. 64

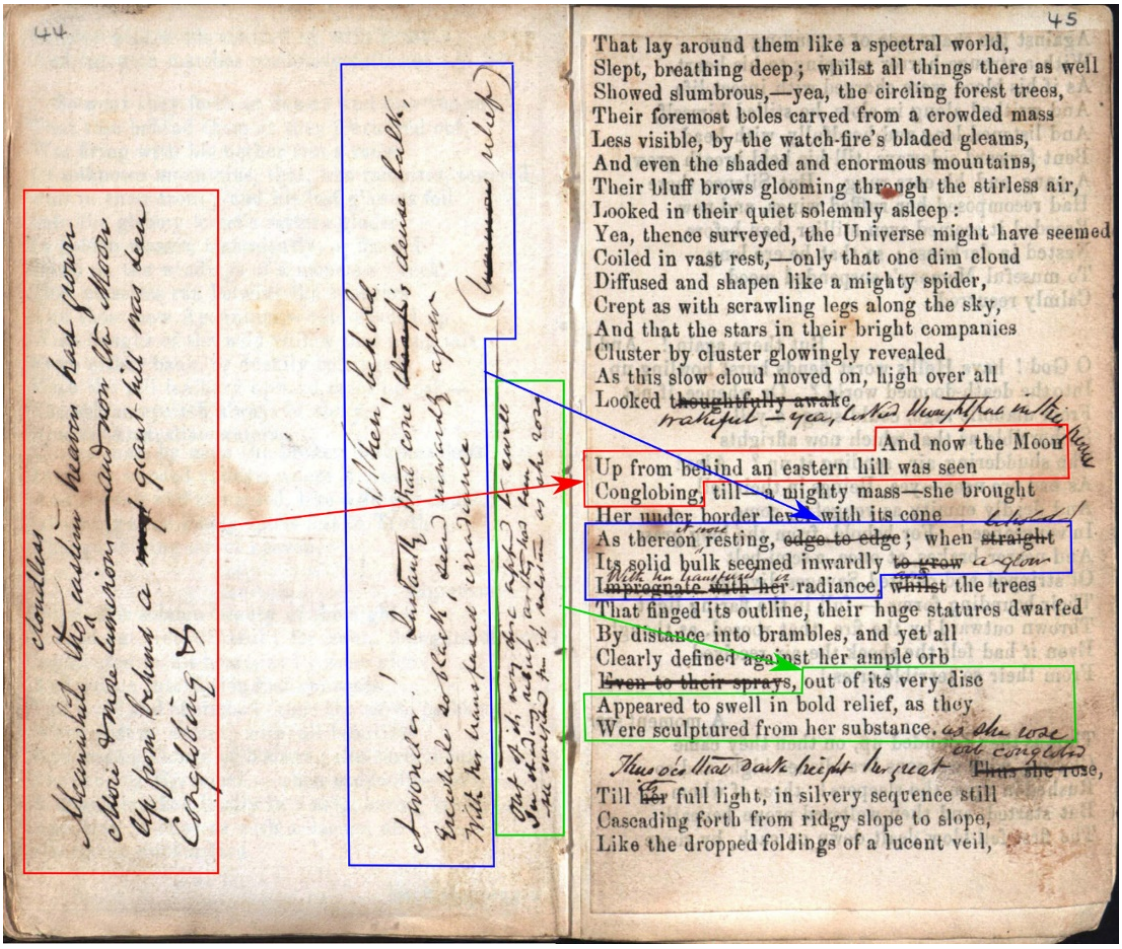

Figure 10.

The temporal sequence of local changes is usually recoverable. Authors provide many cues such as the position of inserted text, deletion marks, sense, the carrying over of changes into revised blocks, etc. that make this possible, even in extreme cases like that shown in Figure 10. A survey of 1,247 local revision-sites in the manuscripts of Harpur revealed that the temporal sequence of changes could be recovered with a high degree of probability in 99.45% of cases. In the doubtful cases, while the layering approach does require the assignment of variants to specific layers, there is no comparative disadvantage. This is because the embedded variant method also requires the assignment of variants to an explicit sequence or nesting of markup. 65

Writing software to reliably process transcriptions coded for the graphical layout of text into a coherent sequence of temporal layers is ‘challenging’, even for a single project. Such software requires the transcription either to be weighed down with extra encoding in addition to the already heavy markup needed to describe the text’s graphical layout [Muñoz 2015], or to be encoded twice: once for the graphical layout and once to describe the temporal or “linear” sequence [Rehbein 2013] [Van Hulle 2015]. The editing of transcriptions and their subsequent reuse are thus impaired [Muñoz 2015] [Brüning 2013]. 66

It looks then as if embedding variants directly into the text-stream at each revision-site leads to serious and insurmountable problems, which can only be resolved by *not* embedding variants into the transcription in the first place. What the layered approach offers instead is a far simpler data model that has no need of customised software to make text readable, searchable and comparable — for it already is. 67

3.3 Ease of Editing

The model of versions and layers also has other advantages. Since the text no longer contains complex markup to describe internal variation it can be edited easily. The only remaining textual features that need to be represented via 68

inline markup are fairly simple textual formats at the paragraph or character levels. Having a readable text mostly devoid of complex markup to compare with the facsimile of the original speeds up initial transcription too. Although as many as nine layers may be required for complex cases, this is mostly confined to a few heavily revised poetical texts. For longer works, such as chapters of a novel, one to three layers should suffice.

Layers are not definitive, and like all text files can be easily changed after initial transcription by the human editor to reflect changing interpretations of the text or increased knowledge of the author's habits. Layers offer an alternative way of organising the same data as inline markup seeks to encode but in a far simpler form. 69

3.4 Longevity

Another key advantage is longevity. The representation of versions and layers generated by the model requires only Unicode text, than which, as already pointed out, there is nothing more stable in the digital world. 70

Although text is one of the most compact forms of data, each layer requires a complete copy of the document, including its local changes. In most cases this is limited to two or three layers. In the case of Charles Harpur, one poem, "Sir Gilbert Blount", required nine layers [Harpur 1865-67, 68–69]. Apart from a few cases like this, out a corpus of 4,596 manuscript pages, the average number of layers needed per manuscript version was only 2.1. Modern computers have many gigabytes of storage to accommodate movies, images and sound files that are many times larger than such simple text files as these. 71

Layers can be grouped within folders or directories, one folder for each version, and one over-arching folder for each work containing the versions. The use of file directories likewise goes back to Multics in the 1960s and hence has a similar pedigree to ASCII. Like Unicode, it can be assumed it will continue to exist into the distant future. 72

4. Disadvantages of versions and layers

The main disadvantage of the layering and versioning approach is, at present, its unfamiliarity. Current editors of modern manuscripts, who are used to embedding variants in the transcription as they encounter them, would need to adjust their practice. Technicians who process the transcriptions would also need to learn new, albeit simpler, techniques. To be adapted to the new approach existing texts would have to be converted by splitting them into layers. Although practicable, this would involve some customisation of the conversion software to accommodate different encoding conventions, as well as manual intervention in difficult cases of textual variation. These costs would not be insignificant, but they would all be short-term, and outweighed by the longer-term benefits. For new projects using layers from the start these problems would not exist. 73

The computation of differences between versions and layers is sometimes imperfect. For example, in the text of Figure 11 [Harpur 1836-68, 410]: "Nought might reward ~~the his~~ the ~~aim~~ blow, however true;" a difference tool would compute the second "the" to be shared by layers 1 and 3, whereas in fact it was deleted and then restored. 74

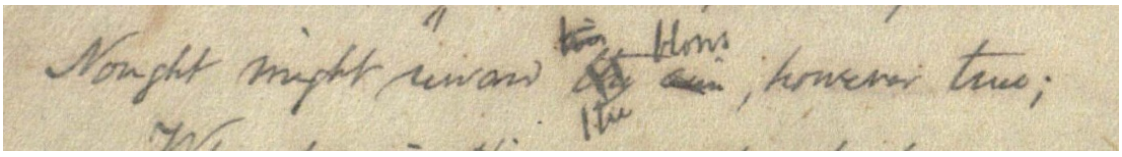

Figure 11.

In the Harpur manuscripts about 2% of inline variants are restorations of earlier readings. Although this detail is not recorded in the layer representation itself it will still be visible in the facsimile, and if the page image is connected closely with the text this can easily be checked by the reader or editor. Although small inaccuracies such as these may be inherent in the layer model they are very much dwarfed by the widely recognised inaccuracies and limitations of hierarchical markup systems for representing internal variation [Pierazzo 2014b, 131] [Fiormonte 2010] [Neyt 2006]. Heavily revised holographs are also extremely difficult to record using embedded markup, because the encoding quickly 75

becomes too complex, but this difficulty is not shared by the layering technique, where the text of each layer is always simple, however complex the revisions. Variants are also not allowed to overlap when using embedded hierarchical markup, but with layering they are. This is a significant advantage.

As a consequence of delegating to a piece of software the task of computing differences between versions and layers, the accuracy of that process is obviously determined by the efficiency of the software. While no program based on a heuristic algorithm can be perfect, the degree of error introduced in this way can be reduced by subsequent versions of the software, or by new and better programs, while retaining exactly the same versions and layers. Biologists have progressively improved both the speed and accuracy of their sequence alignment programs since the 1970s. Adopting this new strategy of organising textual data will encourage the same process of progressive improvement to play out in textual archiving and editing. 76

5. Implementation

Although the model of versions and layers presented here greatly simplifies the encoding of the textual content of documents, it does not reduce it to plain text. For example, what is to be done about features like underlining, semantic metadata, print formatting like italics, divisions into stanzas, paragraphs, speeches, page numbers etc.? Since the concept of versions and layers is unrelated to the technology used to represent them, each layer may be represented in HTML, the markup language of the Web, or digital humanists may prefer XML, the eXtensible Markup Language, or LMNL the layered markup notation language [Piez 2014]. Alternatively, and for greatest longevity, a standoff representation where all external annotation and feature markup is kept separate from the text will insulate it from the rapidly changing technological standards that govern such markup and annotation technologies [Schmidt 2016a]. Using this approach all external annotation may be stored in the form of a layer of markup corresponding to a single layer of text, or using any other desired system. Such markup systems will probably be replaced by something else in the future. This may lead to data loss, but with the standoff representation the text at least will remain coherently readable, searchable and reusable. 77

The preparation of versions and layers can be accomplished using a simple plain text editor. The transcriber may display the source document as an image on one side of the computer screen, and copy the transcribable text on the right, starting with the text's final state. Once a complete copy of this has been obtained it may be regarded as sufficient in itself. If not, further layers can be transcribed by copying the final layer, and then undoing the topmost change at each local revision-site to record, successively, the lower layers. If the deleted text in some layer is unreadable due to being obscured by overwriting or crossing-out, such cases can be dealt via markup (internal or external) to indicate unclear or missing text. 78

Embedded mathematical formulae and graphics, as found for example in the Wittgenstein manuscripts [Pichler 2016], can be treated similarly. An approximation of the formula or graphic in plain text can be used to serve as a placeholder. For example, the formula x^2 could be written "x^2" or an image as "[picture of a ship]". This placeholder can then be marked up externally with the actual formula or graphic, and rendered using appropriate software into more elegant forms that replace the placeholder with the actual formula or graphic. In this way all content, textual or otherwise, can be recorded successfully in layers. 79

5.1 Comparison of versions and layers

The final piece of technology needed to make this model work is the ability to compute differences. Since the versions and layers are plain texts (or, if marked-up may be recovered as plain texts) they can simply be compared with each other. Such software is available now and will be in the future, since computation of differences is a fundamental computing problem. For example, computer scientists use difference-computing tools to save revisions of computer code for later restoration. Difference-computing software can evolve over time without affecting the simple representation of the textual data as versions and layers. [5] 80

The display of versions and layers can be accomplished by using comparison software to compute the differences 81

between layers, or between versions. When comparing layers the shared text may be greyed-out as a reminder to the reader-user that the text does not have the status of a full version. Full versions may have their text displayed in black. An example where this has been done can be seen in the edition of Harpur's multi-versioned poem "Creek of the Four Graves" [Eggert 2019]. The display of both versions and layers employs red (to indicate deletion) and blue (to indicate insertion).

6. Conclusion

If we are to make any significant progress in transcribing the vast collection of manuscript and early-print material in our libraries and museums, a much more efficient and durable method of recording their content than embedding internal variation data into the text is needed. The latter established method requires us to revise the transcriptions we already have and the software that manipulates them as technology continues to evolve. This prospect is unviable, and a change in our practice is needed. The present model of versions and layers provides a justification for change.

Notes

[1] [1.] i.e. transpositions, connected variants, open variants, incoherent corrections, separately revised test-blocks, and restoration of previously deleted text.

[2] [2.] Scholarly editors typically use the term 'state' to indicate a document- or version-wide text. The technical requirement addressed here suggests its adaptation for local sites of textual revision. In fact this parallels, theoretically if not practically, the detection by bibliographers from the 1940s of in-press correction during the printing of Shakespeare's First Folio. Printed sheets were not discarded, leading to variation in states of the text among copies made during the same print run. In the handpress period, no two surviving copies of the same edition of a printed work need necessarily bear the same state of the text.

[3] [3.]While keyword search will retrieve all documents containing all or most of the words in a search expression but in any order, literal search is more useful for finding specific expressions. A survey by the author of 30 digital scholarly editions revealed that only 10% of editions could find literal expressions that span inline substitutions [Schmidt 2016b].

[4] [4.]Dekker et al. 2015 attempt to overcome this problem by treating internal variants either as formats, or by marking them up as variant clusters to be passed through the collation engine without change. However, as yet they have offered no mathematical model that might vindicate this approach as a general solution for comparing texts containing embedded variants. Schäuble and Gabler (2018) achieved a partial success by extracting the first and last layers of document transcriptions containing embedded variants, and then used a collation tool to compute the differences between versions which were then used to create an intermediary transcription between documents [Schäuble 2018, 167]. The "first" layer includes *currente calamo* corrections [Schäuble 2018, 170], both deletions and their replacements.

[5] [5.]While striving to keep the argument on a purely theoretical footing, some examples of suitable textual alignment tools would establish the practicality of the alignment process. Bourdaillet and Ganascia (2006) with MEDITE were the first to show that Ukkonen's suffix tree algorithm (1995) could be applied to humanities texts for the detection of transpositions [Bourdaillet 2006]. Schäuble and Gabler (2018) used the TUSTEP collate tool to align versions of Virginia Woolf's "A Sketch of the Past" [Schäuble 2018]. The author's own program, nmerge (2009), has been used to align the texts in the Charles Harpur Critical Archive [Eggert 2019].

Works Cited

Bauman 2011 Bauman, S 2011. "Interchange vs. Interoperability", Proceedings of Balisage: The Markup Conference 2011. Montréal, Canada. 2–5 August 2011. Available from: https://www.balisage.net/Proceedings/vol7/html/Bauman01/BalisageVol7-Bauman01.html. [11 January 2019].

Bohnenkamp 2016 Bohnenkamp, A., Henke, S., and Jannidis, F 2016. *Johann Wolfgang Goethe: Faust Historisch-kritische Edition*. http://beta.faustedition.net/. [11 January 2019].

Bourdaillet 2006 Bourdaillet, J. and Ganascia, J.P 2006. “MEDITE: A Unilingual Textual Aligner”, Advances in Natural Language Processing: 5th International Conference on NLP, FinTAL, Turku, Finland, 23-25 August 2006 Proceedings, pp. 458-469. Available from: https://www.researchgate.net/publication/221314746_MEDITE_A_Unilingual_Textual_Aligner. [11 January 2019].

Bryant 2002 Bryant, J 2002. *The Fluid Text A Theory of Revision and Editing for Book and Screen*. University of Michigan Press, Ann Arbor (2002).

Bryant 2008 Bryant, J 2008. *Melville Unfolding. Sexuality, Politics, and the Versions of Typee*. University of Michigan Press, Ann Arbor (2008).

Bryant 2009 Bryant, J 2009. “Herman Melville’s Typee A Fluid Text Edition”. Available from: http://rotunda.upress.virginia.edu/melville/. [11 January 2019].

Brüning 2013 Brüning, G., Henzel, K. and Pravida, D 2013. “Multiple Encoding in Genetic Editions: The Case of ‘Faust’”, *Journal of the Text Encoding Initiative* 4. Available from: http://journals.openedition.org/jtei/697. [11 January 2019].

Dekker 2015 Dekker, R.H.,Van Hulle, D., Middell, G., Neyt V., and Van Zundert, J 2015. “Computer-supported collation of modern manuscripts: CollateX and the Beckett Digital Manuscript Project”, *Digital Scholarship in the Humanities* 30.3 (2015): 452–470.

Durusau 2006 Durusau, P 2006. “Why and how to document your markup choices”. In L. Burnard, K. O’Brien O’Keefe, and J. Unsworth (eds), *Electronic Textual Editing*, New York: MLA, pp. 299–309.

Eggert 2019 Eggert, P 2019. “Charles Harpur Critical Archive”. Available from: http://charles-harpur.org/View/Compare/?docid=english/harpur/poems/h080&version1=/h080a/layer-final [11 January 2019].

Fiormonte 2010 Fiormonte, D., Matiradonna, V. and Schmidt, D 2010. “Digital Encoding as a Hermeneutic and Semiotic Act: The Case of Valerio Magrelli”, *Digital Humanities Quarterly* 4.1. Available from: http://www.digitalhumanities.org/dhq/vol/4/1/000082/000082.html. [11 January 2019].

Fraistat 2018 Fraistat, N., Denlinger, E. and Viglianti, R 2018. (n.d.). “The Shelley-Godwin Archive”. Available from: http://shelleygodwinarchive.org/. [11 January 2019].

Gabler Gabler, H. W. (n.d.). “Textual Criticism and Theory in Modern German Editing”. http://www.academia.edu/856505/Textual_Criticism_and_Theory_in_Modern_German_Editing. [11 January 2019].

Gabler 1986 Gabler, H. W 1986. “Afterword”. In *James Joyce Ulysses The Corrected Text*. Bodley Head, London (1986), pp. 647–650.

Greetham 1994 Greetham, D 1994. *Textual Scholarship: An Introduction*. Garland, New York and London (1994).

Göbel 2015 Göbel, M 2015. “Das TextGrid Laboratory: Zehn Jahre Software-Entwicklung”. In Neuroth, H., Rapp, A. and Söring S. (eds), *TextGrid: Von der Community – für die Community Eine Virtuelle Forschungsumgebung für die Geisteswissenschaften*. Werner Hülsbusch, Glückstadt (2015), pp. 251–259.

Harpur 1836-68 Harpur, C 1836–1868. “A87-2” Available from: http://charles-harpur.org/View/Pages/?docid=english/harpur/A87-2. [11 January 2019].

Harpur 1849-63 Harpur, C 1849–1863. “C376” Available from: http://charles-harpur.org/View/Pages/?docid=english/harpur/C376. [11 January 2019].

Harpur 1853a Harpur, C 1853a. *The Bushranger: A Play in Five Acts and Other Poems*. W. R. Piddington, Sydney (1853). Available from: http://charles-harpur.org/View/Pages/?docid=english/harpur/A87-2.

Harpur 1853b Harpur, C 1853b. “Trust in God”, *Empire*, 20 June 1853, p. 2410. Available from: http://charles-harpur.org/corpix/english/harpur/EMP/20-JUN-1853-P2410-H646A.jpg. [11 January 2019].

Harpur 1855 Harpur, C 1855. “B78”. http://charles-harpur.org/View/Pages/?docid=english/harpur/B78. [11 January 2019].

Harpur 1855-62 Harpur, C 1855–1862. “C384”. Available from: http://charles-harpur.org/View/Pages/?docid=english/harpur/C384. [11 January 2019].

Harpur 1859 Harpur, C 1859. “Trust in God”, *Australian Home Companion* IV, 5 November 1859, p. 467. Available from: http://trove.nla.gov.au/newspaper/article/72486054. [11 January 2019].

Harpur 1863 Harpur, C 1863. “A89”. Available from: http://charles-harpur.org/View/Pages/?docid=english/harpur/A89. [11 January 2019].

Harpur 1865-67 Harpur, C 1865-1867. “A95” Available from: http://charles-harpur.org/View/Pages/?docid=english/harpur/A95. [11 January 2019].

Heckel 1978 Heckel, P 1978. “A technique for isolating differences between files”, *Communications of the ACM* 21.4: 264–268.

Huitfeldt 1993 Huitfeldt, C 1993. “The Wittgenstein Archives at the University of Bergen: Project Report 1990-1993 and Critical Evaluation”. Available from: http://wab.uib.no/1990-99/reports/no9.htm. [11 January 2019].

Jebb 1905 Jebb, R. C 1905. *Bacchylides: The Poems and Fragments*. Cambridge University Press, Cambridge (1905).

Jenab 2014 Jenab, K., Noori, K., Weinsier, P. D. and Khoury, S 2014. “A dynamic model for hardware/software obsolescence”. *International Journal of Quality & Reliability Management*, 31.5: 588–600.

Juxta 2013 Juxta Commons 2013. “Advanced Options for XML Texts” Available from: http://juxtacommons.org/guide#advanced. [11 January 2019].

Muñoz 2015 Muñoz, T., and Viglianti, R 2015. “Texts and Documents: New Challenges for TEI Interchange and Lessons from the Shelley-Godwin Archive”, *Journal of the Text Encoding Initiative* 8. Available from: http://journals.openedition.org/jtei/1270. [11 January 2019].

Myers 1987 Myers, E. W 1987. “An O(ND) difference algorithm and its variations”, *Algorithmica*, 1: 251–266.

NLA 2016 NLA 2016. National Library of Australia. “Collection Statistics”. Available from: https://www.nla.gov.au/collections/statistics. [11 January 2019].

NLA 2017 NLA 2017. “Preparing for copyright term changes in 2019”. Available from: https://www.nsla.org.au/sites/default/files/documents/nsla.copyright-preparing-changes-2019.pdf. [11 January 2019].

Nedo 1993 Nedo, M 1993. *Ludwig Wittgenstein Wiener Ausgabe, Einführung/Introduction*. Springer, Vienna, New York (1993).

Neyt 2006 Neyt, V 2006. “Fretful Tags Amid the Verbiage: Issues in the Representation of Modern Manuscript Material”, *Literary and Linguistic Computing* 21, supplement: 99–111.

Pauly 2003 Brill’s New Pauly 2003. *Encyclopedia of the Ancient World*. Brill, Leiden (2003).

Pichler 2016 Pichler, A 2016. “Wittgenstein Source”. Available from: http://wittgensteinsource.org/. [11 January 2019].

Pierazzo 2009 Pierazzo, E 2009. “Digital genetic editions: The encoding of time in manuscript transcription”. In M. Deegan and K. Sutherland (eds), *Text Editing, Print and the Digital World*. Ashgate, Farnham (2009), pp. 169–186.

Pierazzo 2014a Pierazzo, E 2014a. “Digital Documentary Editions and the Others”, *Scholarly Editing* 35. Available from: http://www.scholarlyediting.org/2014/essays/essay.pierazzo.html. [11 January 2019].

Pierazzo 2014b Pierazzo, E 2014b. “Digital Scholarly Editing: Theories, Models and Methods”, London: Routledge. Available from: http://hal.univ-grenoble-alpes.fr/hal-01182162. [11 January 2019].

Piez 2014 Piez, W 2014. “TEI in LMNL”, *Journal of the Text Encoding Initiative* 8 (2014). Available from: http://journals.openedition.org/jtei/1337. [11 January 2019].

Rehbein 2013 Rehbein, M. and Gabler, H. W 2013. “On Reading Environments for Genetic Editions”, *Scholarly and Research Communication* 4.3: 1–21. Available from: http://src-online.ca/index.php/src/article/viewFile/123/260. [11 January 2019].

Saller 2003 Saller, H 2003. HMNL “The HyperNietzsche Markup Language”, *Jahrbuch für Computerphilologie* 5. Available from: http://computerphilologie.digital-humanities.de/jg03/saller.html. [11 January 2019].

Sandborn 2012 Sandborn, P 2012. “Software Obsolescence”. In B. Bartels, U. Ermel, M. Pecht, P. Sandborn (eds), *Wiley Series in Systems Engineering and Management, Strategies to the Prediction, Mitigation and Management of Product Obsolescence*. Wiley, Hoboken, NJ (2012).

Sattler 1979 Sattler, D. E 1979. “Marginalien zur Frankfurter Hölderlin-Ausgabe”, *MLN* 94.3: 601–606. Available from: http://www.jstor.org/stable/2906534. [11 January 2019].

Schmidt 2009 Schmidt, D 2009. “Merging Multi-Version Texts: a Generic Solution to the Overlap Problem”. In *Proceedings of Balisage: The Markup Conference 2009. Balisage Series on Markup Technologies*, 3. Available from: http://doi.org/10.4242/BalisageVol3.Schmidt01. [11 January 2019].

Schmidt 2014 Schmidt, D 2014. “Towards an Interoperable Digital Scholarly Edition”, *Journal of the Text Encoding Initiative* 7. Available from: https://journals.openedition.org/jtei/979. [11 January 2019].

Schmidt 2016a Schmidt, D 2016a. “Using standoff properties for marking-up historical documents in the humanities”, *it – Information Technology* 58.2 (2016): 63–69.

Schmidt 2016b Schmidt, D 2016b. “Enhancing Search for complex historical texts”. In *Proceedings of Digital Humanities Australasia 2016: Working with Complexity, Hobart 20–23 June 2016* Available from: http://charles-harpur.org/presentations/dha2016.search.pdf. [11 January 2019].

Schäuble 2018 Schäuble, J. and Gabler, H.W 2018. “Encodings and Visualisations of Text Processes across Document Borders”. In R. Bleier, M. Bürgermeister, H. W. Klug, F. Neuber, G. Schneider (eds), *Digital Scholarly Editions as Interfaces*, BoD: Norderstedt (2018), pp. 165–191.

Shillingsburg 1996 Shillingsburg, P 1996. *Scholarly Editing in the Computer Age Theory and Practice Third Edition*. University of Michigan, Ann Arbor (1996).

Sperberg-McQueen 2000 Sperberg-McQueen, C. M., Huitfeldt, C. and Renear, A 2000. “Meaning and Interpretation of Markup —not as simple as you think”. In *Acoustics, Speech, and Signal Processing Newsletter, IEEE* 2.3 (2000): pp. 215–234.

TEI 2017 TEI Consortium 2017. “Guidelines for Electronic Text Encoding and Interchange”. Available from: http://www.tei-c.org/P5/. [11 January 2019].

Tagore Tagore, R (n.d.). “Bichitra: Online Tagore Variorum: School of Cultural Texts and Records”. Available from: http://bichitra.jdvu.ac.in/bichitra_user_manual.php. [11 January 2019].

Ukkonen 1995 Ukkonen, E 1995. “Online construction of suffix trees”, *Algorithmica* 14.3 (1995): 249–260. Available from: http://www.cs.helsinki.fi/u/ukkonen/SuffixT1withFigs.pdf. [11 January 2019].

Van Hulle 2011 Van Hulle, D 2011. “Modern Manuscripts and Textual Epigenetics: Samuel Beckett’s Works between Completion and Incompletion”, *Modernism/modernity* 18.4 (2011).

Van Hulle 2015 Van Hulle, D., and Neyt, V 2015. “Samuel Beckett Digital Manuscript Project”. Available from: http://www.beckettarchive.org. [11 January 2019].