2019

The Charles Harpur Critical Archive: A History and Technical Report

Paul Eggert
peggert@luc.edu

Desmond A. Schmidt
Queensland University of Technology, desmond.allan.schmidt@gmail.com

THE CHARLES HARPUR CRITICAL ARCHIVE

A HISTORY AND TECHNICAL REPORT

By Desmond Schmidt (Brisbane) and Paul Eggert (Chicago)

This report is a preprint version before final corrections. The published version (*International Journal of Digital Humanities*, vol. 1, 2019) should be consulted for quotation purposes.

Abstract:

This is a history of and a technical report on the Charles Harpur Critical Archive (CHCA), in preparation since 2009. Harpur was a predominantly newspaper poet in colonial New South Wales from the 1830s to the 1860s. Approximately 2700 versions of his 700 poems in newspaper and manuscript form have been recovered. In order to manage the complexity of his often heavily revised manuscripts traditional encoding in XML–TEI, with its known difficulties in handling overlapping structures and complex revisions, was rejected. Instead, the transcriptions were split into simplified versions and layers of revision. Markup describing textual formats was stored externally using properties that may freely overlap. Both markup and the versions and layers were merged into multi-version documents (MVDs) to facilitate later comparison, editing and searching. This reorganisation is generic in design and should be reusable in other editorial projects.

Keywords:

Charles Harpur, Romanticism, digital archive, multi-version documents (MVDs), XML–TEI, standoff markup

1. Background

This is a history and a technical report on the Charles Harpur Critical Archive (CHCA). Looking back to its beginnings in 2009, it is clear that we did not fully appreciate the size and nature of the challenge that lay ahead. Funding from the Australian Research Council paid, initially, for data input and research assistance. The development of the technical side only came later: two false starts followed by the one we have now settled upon. The development site is at <charles-harpur.org>.[1] Formal publication is planned for mid 2018, the 150th anniversary of Harpur's

1 Paul Eggert is project leader and editor of the CHCA (2009–). Desmond Schmidt is the technical lead and programmer (2013–), and Meredith Sherlock served as digital archivist 2009–2016. Other principal contributors include Elizabeth Webby (annotations), Chris Vening

death. This archival expression of the project will lay the groundwork for more properly editorial and other interpretative endeavours, firstly by Paul Eggert as editor and as soon as practicable by collaborators.

Romanticists and others reading this report are unlikely even to have heard of Harpur, who was born in 1813. Yet we believe that assessment of his works, once they are available in a properly contextualised form, will lead many to form the view that he was a major talent who, for book-historical and other reasons, has been almost entirely lost from view. Living and writing in colonial New South Wales from the early 1830s until his death in 1868 Harpur cut a considerable figure locally: his verse made at least 900 appearances in the colonial and intercolonial press. But publishing opportunities for him other than in newspapers were almost non-existent.

He maintained clippings files and retained some manuscript copies from the start, but he yearned for a magisterial volume of his collected verse to appear in London. He prepared the way for it, late in his life, by copying, sequencing and then, in successive manuscript books, revising and re-sequencing his poems. We have recovered approximately 2700 versions of his 700 poems.[2] In addition, many have extensive authorial notes in which Harpur sought to key his newspaper poetry into the moving agendas of the day. The complexities of this evidence, together with the disarranged state of his manuscripts in the Mitchell Library in Sydney, defeated or compromised a series of editorial projects from the 1940s in Australia that aimed to capture the full range of his poetic achievement in book form (see Eggert 2016).

2. Method

The preparation of the CHCA has been an arduous journey, replete with unforeseen problems. By documenting the process in its entirety it is hoped that other editorial projects may learn from our mistakes and successes alike. A more detailed discussion of the theoretical implications of the concepts and methods that were finally adopted appear in a separate publication.[3]

Problems of encoding in XML–TEI

The process of preparing the CHCA began in 2009. High-quality scans were provided by the Mitchell Library, State Library of New South Wales, Sydney. Skilled inputters – MA students in the School of Cultural Records at Jadavpur University in Kolkata – were employed to transcribe the 24 manuscript books using simplified XML-TEI codes. Their job was to code, quickly, what they saw in front of them, in other words to transcribe documents as a series of scribal acts, to ignore problems of work-and-version differentiation, and to leave the encoding of problematic deletions and additions for later resolution. Their efforts provided an invaluable first-cut

(biographical entries) and Michael Falk (qualitative assessments of poems). Loyola University Chicago has provided funding since 2015.

2 The cut-off date for the project is 1900. The 2700 items include the newspaper and manuscript forms, and a heavily bowdlerised collection in book form arranged by his widow fifteen years after his death. Its texts were the source of future anthologies and most Harpur collections until the 1980s.

3 See Eggert and Schmidt (2018). Unavoidably there is overlap between the two reports, but their differing emphases have warranted separate treatments.

transcription of complex and often hard-to-read manuscripts. The resulting files were carefully checked and revised, in part-time work over several years, by Meredith Sherlock along with other duties, notably the transcription of the newspaper cuttings and other printed forms. She also organised the document transcriptions into provisional work-and-version designations and maintained bibliographical control of the files until it was superseded by the website's content management system.[4]

TEI encoding is complex to begin with but, once mastered, remains unavoidably subjective. We found that our early TEI encoding decisions changed as we gained fuller experience of the manuscripts. Yet the task never seemed to get any simpler. In practice, the encoding requirements of a large body of texts, especially if they are manuscripts, are not fully known until the work is finished. Inconsistency creeps in. The encoding of revision sites (deletions and additions) is especially problematic, since the tools to process the encoded files will be later developed by technicians who do not share the transcribers' shifting understanding of the meaning and applicability of the codes.

So it was for the CHCA. There was no satisfactory off-the-shelf solution available. Various experiments were undertaken, first using the already existing platform Heurist, which had been adapted to deal with XML-TEI input, and subsequently using the newly developed archival, tools, storage and workflow system called AustESE.[5] Yet both approaches raised fresh problems, which were not easy to address.

These difficulties prompted a bold simplification and reorganisation of the encoded transcriptions. This was in part a response to the complexities of inline TEI markup, which, despite utopian claims made for it in its early days, makes interoperability achievable only within but not across projects.[6] It was also based on the realisation that developer and tool-vendor interest in XML, the language in which TEI encoding is expressed, has been rapidly diminishing (Figure 1), leaving the digital textual-studies community potentially vulnerable.[7] It seemed to us doubtful that commercial and open-source developers would continue in future to support XML

4 Sherlock was guided by (but also corrected as necessary) Holt and Perkins (2002).

5 Intended for archives and scholarly editions, only remnants of it on the Web remain. It was hacked in 2015, effectively destroyed, for there has been no funding available to reinstate it. It was a complicated system that addressed a wide variety of use-cases. It used Drupal as a content management system (CMS) and could deal easily with XML-TEI, although it was in fact agnostic as to the language used. The CHCA now uses a CMS of its own.

6 Discussed in Eggert 2018a. For a summary of the Ecdosis tools used in the CHCA see Eggert 2018b (About→Technical Design).

7 The graph in Figure 1 charts the declared response format of 17,103 public APIs (application programming interfaces) of Web services listed in the Programmable Web index (https://www.programmableweb.com/category/all/apis), accessed 6 January 2018). See also Patrizio 2016 and the url for his graph at <http://images.techhive.com/images/article/2016/06/stackoverflowqs-100665502-orig.png>: both accessed 9 January 2018.

or domain-specific vocabularies of it such as TEI when they appear to have been losing interest in XML since about 2008 (Figure 2)[8].

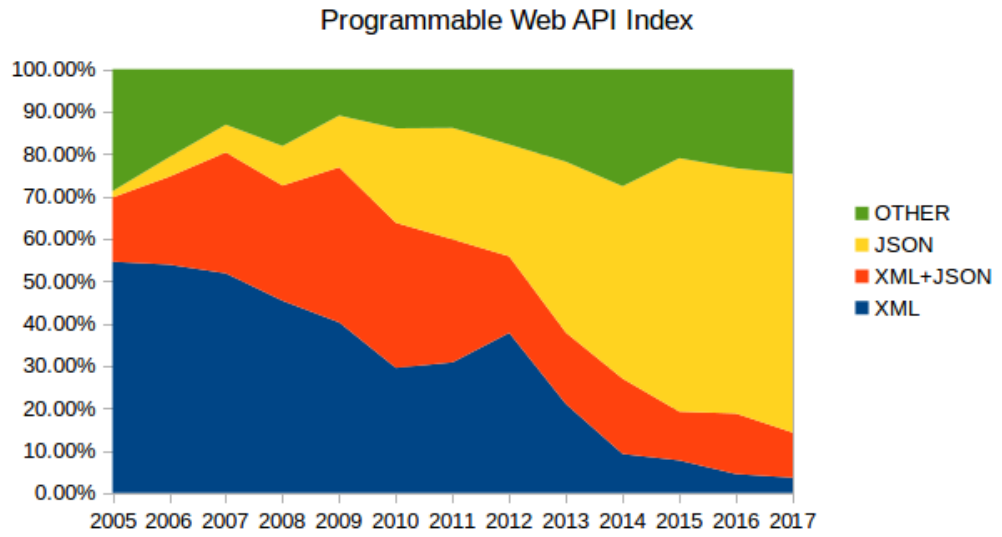

Figure 1: XML vs JSON APIs for public web services 2005-2017

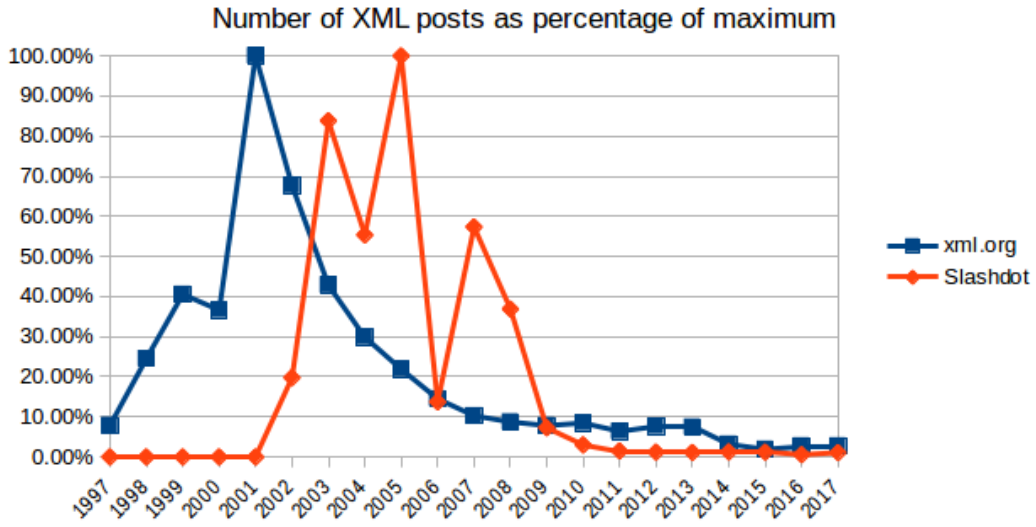

Figure 2: Popularity of XML as a topic on Xml.org and Slashdot

8 The xml.org mailing list charts the popularity of XML from its inception in 1997 to the present day. The popular and long-standing slashdot.org news site reports on technical issues including XML from 2001. Since the popularity of Slashdot varied year on year the actual numbers of XML posts have been normalised by expressing them as percentages of the total posts for each year. The values for both Xml.org and Slashdot were expressed as percentages of their most popular year to make them directly comparable.

Lou Burnard (2013) has argued that the TEI is not tied to XML, and that therefore the Guidelines could in future be redefined using a new markup formalism. But he does not say what that might be. There seem to be only a few possibilities. The TEI Guidelines specify a custom markup syntax, which would be ill-suited to a fixed-syntax language like HTML. Although the current structure of TEI could be mapped onto JSON (JavaScript Object Notation), JSON is not a mixed-content format that would allow text and tags to be combined, nor is it easily user-typeable. It is designed for server-to-client communications within Web applications, rather than as a textual document format. Other than these two possibilities any other custom technology that may be developed as a replacement for XML would tend to isolate the TEI community from mainstream Web development. Whether there is a confident way forward into the future for XML-TEI remains an open question. To us, for now, it seems doubtful.

The MVD format

XML requires an ordered hierarchy of content objects, a tree structure. As every transcriber soon learns, literary texts are neither orderly nor hierarchical. Elements frequently overlap, and workarounds have to be resorted to. Each workaround makes it harder for the encoded text to be subsequently processed by external tools. Deletions and additions, for example, are normally encoded in TEI as if they are formats of a single linear text when they are actually variant readings that can only be read in parallel.

The Harpur Archive has pioneered a simpler route that is less dependent on current technical approaches. The XML files already prepared by the transcribers were first simplified and made syntactically uniform. Even then, significant semantic variation remained because of the ways in which the permitted tags had been used. This inconsistency reflected our evolving understanding of the texts and the variety of people who had encoded them.

The XML files of each documentary version were then split into layers, each of which was a more or less readable transcription organised around the local state of the text at each revision site. (In contrast to this working concept of *layers*, during the archival phase of the project we have understood a *version* to be a document-wide state in which the author left the text at some point in time.[9]) Next, the layers were split into markup and plain text. Finally, the layers of markup and of text were merged into separate multi-version documents (MVDs).

An MVD is a file (i.e. application) format that may be visualised as a variant graph,[10] which merges the separate transcriptions of each layer of a work (Figure 3). It is intuitively easy to understand and has gained some traction in other fields of editing such as biblical studies and in the visualisation of variants. In a variant graph the text held in common is stored only once. Where readings diverge one version has its reading recorded on the top loop and the other's on the bottom loop. Each loop re-enters the main text when the variance of its layer ends.

9 The separate publication mentioned above (Eggert and Schmidt 2018) reflects further on the distinction between layer and version.

10 The MVD in fact simplifies the variant graph, into which it may be readily converted or vice versa.

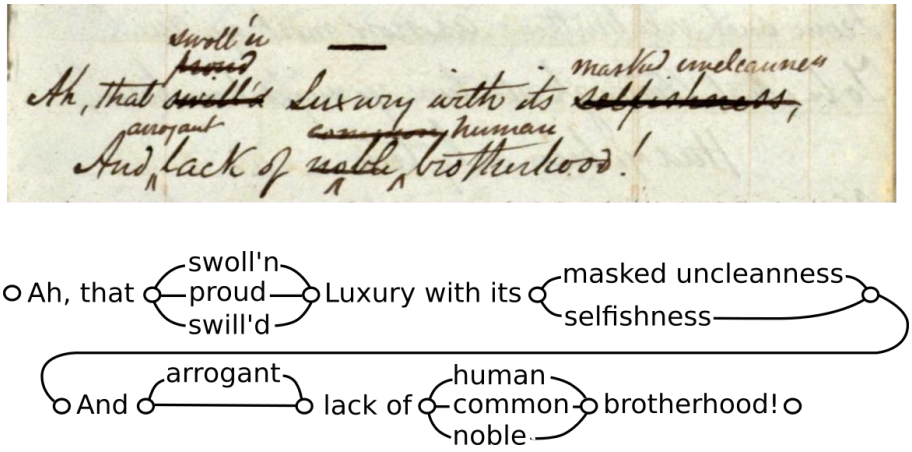

Figure 3. Variant graph based on part of MS B78, p. 89

For Harpur, building an MVD took anything from less than a second for the shorter poems to about a minute for the longest, multiversion poems, such as ‘The Creek of the Four Graves’. Once built, an MVD allows instantaneous comparison between any of the layers or versions of a work, regardless of the size of the text or the number of its versions.

A major difference between the MVD format and XML is that MVD is an application format, not a markup language. Therefore the MVD is not a direct substitute for XML. Rather, its purpose is to store in a compact form every version of a work, and to arrange the text in such a way as to facilitate the kinds of operations typically performed in digital scholarly editions such as comparison between versions.

Although the MVD approach can tolerate the presence of textual formats within the versions and layers, separating the markup from the text simplifies later comparison, searching and processing. Textual properties such as italics, stanza and paragraph codes are stored as sets of range-attributes, one set for each version and layer, which point to the corresponding text inside an MVD. So, for instance, five versions of a work require five versions of properties and five of text. This arrangement of versions and layers and the separation of text and markup eliminates the overlapping hierarchies problem that has dogged XML from the start[11]. Textual variants and their properties can now freely overlap.

Another advantage of this approach is a reduction in the complexity of the transcriptions. A survey of the first 99 poems out of 700 revealed 1,495 revision sites. So the total number may be around 10,000. In many cases the XML needed to capture the details of the revisions was excessively complex and nearly impossible to edit. The use of layering reduced this complexity

11 See Pierazzo on the long-recognised problem of overlapping hierarchies for TEI encoding: ‘the reality is that the problem has no real solution’ (2015, 120).

to a series of simple and readable plain-text files. Only 4,615 files were needed to represent 2,738 versions and their layers, with no significant loss of information from the original XML.[12]

The division of each XML–TEI transcription into layers was performed by a program called the Splitter. However, we knew beforehand that the sequence of alterations at any one revision site would often not have been accurately captured for the purposes of constructing an MVD. This is because the XML–TEI codes used were designed to record the appearance or position of revisions rather than their logical or temporal sequence. Although the simpler cases of internal variation could be handled automatically by assigning deletions and additions to their correct layers, in more complex cases the markup had to be revised manually to ensure that the TEI variant tags could split the text into logical layers. The layers had then to be checked for readability, the original XML file recoded where necessary, and then re-split.

Having obtained readable layers from each version, still containing some XML tags, a second program, the Merger, separated the layers into plain text and markup, and then merged both sets of layers into their MVDs.

The Merger tool also linked the page-images with the text by utilising the existing page-break elements in the XML files. The page images themselves had to be manually sliced into segments corresponding to parts of poems that appeared at the top, middle or foot of each page. In complex cases, sections of vertically written text in the margins or on other pages connected via metamarks had to be identified and manually linked. In all, 2800 part-page images had to be created, in addition to the 6000 full-page images of newspapers, manuscripts and books. This entire reorganisation of images and transcriptions took eight months.

This experience confirmed us in the belief that if we were to start again from scratch we would not use TEI at all. Instead we would use something like the Markdown Editor recently devised for the project. In this Editor, the page image scrolls synchronously alongside the transcription. Layers are accessed from separate tabs. Recording a local change is then reduced to simply editing the appropriate layer. The Editor tool has been used during 2016–2017 to transcribe the extant correspondence of Harpur and his circle to 1900. So far we have not encoded layers of revisions in the letters, but the Editor will allow us, or a collaborator, to do so at any time in the future.

Instead of XML the Editor uses what we call a Minimal Markup Language (MML) resembling Markdown and customised for the class of texts being encoded. Surprisingly few codes are needed once XML requirements are ignored and when one proceeds from the assumption that a facsimile image of the transcribed page will always be visible alongside the transcription. MML codes are used only for editing. For viewing, the text is converted into HTML, which is the standard language of the Web. For storage, it is split into plain text and standoff properties as described above.

12 This is less than two layers per version overall because there are hundreds of newspaper and other printed versions that require only one layer each.

Visualisations

Once the transcription files for any one work had been processed to create the MVDs, various visualisations became possible. We now describe two in detail: the Table and Twinview. For the Compare tool, which displays transcriptions of any two versions with their variants highlighted in colour; for the Timeline, which allows the user to explore biographical, writing and other events; and for genealogical trees and other tools, the reader may consult the Harpur website.

THE TABLE OF VARIANT READINGS is a tabular display of variant and invariant readings. Line-grouping formats (e.g. for stanza, paragraph) are removed to permit text-scrolling, but punctuation remains. The text of each version or layer scrolls horizontally across the window and is under the control of the user. The texts of the various versions and layers are stacked vertically, one beneath the other. Each text-segment has either the same or different text to that of the other versions and layers displayed above and below it. The vertical position of each version or layer in the Table, and the chosen base text, can be altered at will as the editor tests out possible transmission scenarios. At the time of writing all problems have not yet been resolved, but the results so far are promising. It is hoped that this way of displaying variant information will render the traditional textual apparatus unnecessary and thus free up the scholarly editor to concentrate on establishing reading texts and on traditional and non-traditional forms of commentary. (This issue is explored in the separate publication.)

TWINVIEW is a new approach to an old problem: how best to display a page of text next to its source image? The problem is one of alignment. Readers instinctively wish to see the correct portion of the image next to its transcribed section of text. This is achieved in Twinview by first turning the page images into a continuously scrolling series of images and then by scrolling the text and its corresponding page images proportionally. The alignment points are the top and bottom of each page image against the positions in the text where the transcription of the page starts and ends. Since the user may scroll anywhere in the text all other points of alignment are calculated proportionally, based on these fixed points, without any heed paid to what is in the page image.

One drawback with this approach is that for any page containing only a portion of a longer poem the page image has to be sliced into sections. For some heavily revised texts, sections of other pages where revisions happen to have been placed by the author need to be included and if necessary rotated. However, this extra work has not proven onerous. In practice, Twinview works well and has allowed accurate alignment between many thousands of page images and their corresponding text with relatively little effort. It is able to handle layers and versions by swapping the page images on demand, and also has a zoom function to show more detail in the images. The usual ‘diplomatic’ formatting of the text incorporating inline changes such as insertions and deletions has been replaced by layers, which can be activated by clicking on tabs.

Deletions and insertions between layers are highlighted in colour: red for deletions and blue for insertions. The same is true for versions. When a layer is displayed, text held in common between layers is greyed-out to indicate its provisional status. When versions are displayed shared text appears conventionally in black. We have found that this approach renders even the most complexly revised texts quite readable.

3. Discussion

The innovations of the CHCA described above raise some important theoretical points that require fuller treatment. For example, the splitting of variants between layers and versions and their storage in separate files has many advantages, but is it justified, and what are the drawbacks? Also exactly what is meant by the distinction between ‘layer’ and ‘version’?

The temporal sequence of independent alterations on a manuscript may, we found, at least in their local context almost always be determined. An author typically leaves many clues: position, content, sense, crossing-out of earlier readings. Carefully considered together, they usually leave little doubt as to the logical sequence of revised readings at any one point. Thus a level 3 change would be an alteration to an alteration of an original reading on the baseline.

But not all changes are independent. Sometimes it will be clear to the transcriber that a revised reading at a particular spot, designated as level 2 because it is the second reading, is linked, whether by handwriting, style, sense or metre, to a revision at a different level elsewhere. The transcriber is free to assign alterations to whatever level seems appropriate so that related revisions will appear together.

A layer is thus best understood as a collection of local readings in their contexts at a particular level as determined by the transcriber or editor. The use of layers to record interim local states does not constitute an editorial claim that they, in their contexts, ever existed as integral texts. They are nevertheless useful because they serve as compact records of a document’s internal variation. They may be compared to one another in order to highlight local changes to the manuscript text as it moved towards the final state in which the author either finished it or abandoned it.

Text that is removed, replaced, inserted or transposed differentiates the layer so formed from its earlier layer. For the purposes of the MVD, there is usually no need to explicitly record these operations per se. This greatly simplifies the markup. If, as sometimes happens, text of a lower layer is left undeleted, either because the author did not settle definitively on the new reading or because of a simple oversight, the undeleted text of the lower layer is labelled explicitly as such so that it can be displayed correctly. One present drawback of this layering technique is that where the author subsequently restores a baseline reading (signalled by its reappearing in a third or higher layer), the text will not show as changed (in the Table or using the Compare display) when the two layers are compared. This occurs, we found, in around five per cent of the revision sites. But the change will nevertheless be visible in the accompanying page image (using Twinview).

As further discussed in the separate publication mentioned above, a version is, strictly, an editorial declaration of a text that, typically, embodies a certain stage in the development of the work. It is derived from a study of the documentary evidence and is usually based on or is an emended form of one or more of those documents. Until that point the texts of documents, archivally captured, are only version candidates. The CHCA has been organised on this

principle, and the statistics given above reflect it.[13] If all alterations are in the same pen and same ink-colour an editor may well confirm that the final state of the text on the autograph, captured archivally, constitutes a version. However, the same manuscript page may bear evidence of two or more versions, belonging to distinct revision campaigns that may be distinguishable by pen or ink-colour.[14] Equally, a page may not be evidence of a new version at all if it is an identical or nearly identical copy of one whose text is already witnessed.

4. Conclusion

Instead of assembling an ever more complex TEI file as a single source to serve all future visualisations and analysis the CHCA has followed the devolved approach of text and external stand-off markup that Phill Berrie pioneered in the late 1990s and developed for a few electronic editions in Canberra, Australia, in the early 2000s.[15] The introduction of the MVD, a radical step forward, has involved a rethinking of this approach. Berrie's standoff markup, which assumed a hierarchical document structure, is now reconceptualised as stand-off properties, which do not. The key advantage achieved by Berrie was the separation of text and markup, which facilitates the reuse of the text and much of its subsequent processing, such as comparison, searching or the handling of hyphenation.[16] The often-voiced objection that standoff markup or properties make the text difficult to edit has proved unfounded. Text and markup in the CHCA are stored separately and are instantly combined for viewing and editing. Saving stores them in standoff format once again, so avoiding the problem altogether.

Charles Harpur himself could never have envisaged the treatment his works have received in the CHCA project, or foreseen the ongoing opportunity for reading and interpretation that its organisation will permit. Sadly, and situated as he was, Harpur himself could not achieve publication of most of his poetry in his lifetime in an enduring and authorial form. Will the publication of the archival expression of the CHCA on the 150th anniversary of his death in 2018 change his prospects? It is hoped that the CHCA, which finally lays all the cards on the table,

13 The statistics may change during the editorial phase of the project, due to start after the formal launch of the CHCA in mid 2018.

14 E.g., 'Trust in God' appears on the inside front cover of A87-1 (Mitchell Library, State Library of New South Wales). Its base layer is a second version of what appears on p. 109 of the same manuscript book. That base layer appeared in *Empire* on 20 June 1853; the second layer with some further development (perhaps via a lost subsequent draft) was printed in *Australian Home Companion* on 5 November 1859. For a theoretical defence of versional editing, see Eggert 2017.

15 See, e.g., The Jerilderie Letter edition at http://asecentre.org/JITM/index.html, accessed 2 November 2017. Click on *Change Settings* then on *Create Perspective*. Just In Time Markup is applied as the perspective is created. Various options are available. The project ceased in 2004 before it had had the benefit of a professional page design; it looks antiquated now.

16 See, e.g., the discussion by Bauman 2016 of the tortuous handling of end-of-line hyphens in XML. In comparison, when there is no markup in the text (in the CHCA, markup is held as standoff properties) the problems Bauman notes as created by using XML are removed.

will encourage ongoing cycles of engagement with his writings well into the future. If so, then his status as a fine, late-Romantic poet and fiery nationalist, forgotten no longer, will at last be recognised.

Desmond Schmidt

Queensland University of Technology

desmond.schmidt@qut.edu.au

Paul Eggert

Loyola University Chicago

pauleggert7@gmail.com

Works Cited

Bauman, Syd (2016). 'The Hard Edges of Soft Hyphens'. Conference paper. Balisage 2016. <http://www.balisage.net/Proceedings/vol17/html/Bauman01/BalisageVol17-Bauman01.html>, accessed 2 November 2017.

Burnard, Lou (2013). 'Resolving the Durand Conundrum'. *Journal of the Text Encoding Initiative*. 6 <http://journals.openedition.org/jtei/842>

Eggert, Paul (2016). 'Charles Harpur: The Editorial Nightmare'. *JASAL*. 16.2. <https://openjournals.library.sydney.edu.au/index.php/JASAL/article/view/11011>, accessed 2 November 2017.

—— (2017). 'Versional Editing of a Romantic Poet'. *Tipofilologia*. 10. 11–24.

—— (2018a). 'The Archival Impulse and the Editorial Impulse', *Variants*. 14, forthcoming.

—— (2018b), ed. *The Charles Harpur Critical Archive.* <http://charles-harpur.org>, accessed 9 January 2018.

——, and Desmond Schmidt (2018). 'Romantic Poetry and Technical Breakthrough: The Charles Harpur Critical Archive'. *Archiv*. 255.1, forthcoming.

Holt, Elizabeth, and Elizabeth Perkins, ed. (2002). *The Poems of Charles Harpur in Manuscript in the Mitchell Library and in Publication in the Nineteenth Century: An Analytical Finding List*. Canberra: Australian Scholarly Editions Centre.

Patrizio, Andy (2016). 'XML Is Toast: Long Live JSON'. *CIO*. 9 June 2016. <http://www.cio.com/article/3082084/web-development/xml-is-toast-long-live-json.html>, accessed 9 January 2018.

Pierazzo, Elena (2015). *Digital Scholarly Editing: Theories, Models and Methods*. Farnham, Surrey: Ashgate.