

Digital Editions and Version Numbering

Paul A. Broyles <pabroyle_at_ncsu_dot_edu>, North Carolina State University

Abstract

Digital editions are easily modified after they are first published — a state of affairs that poses challenges both for long-term scholarly reference and for various forms of electronic distribution and analysis. This article argues that producers of digital editions should assign meaningful version numbers to their editions and update those version numbers with each change, allowing both humans and computers to know when resources have been modified and how significant the changes are. As an examination of versioning practices in the software industry reveals, version numbers are not neutral descriptors but social products intended for use in specific contexts, and the producers of digital editions must consider how version numbers will be used in developing numbering schemes. It may be beneficial to version different parts of an edition separately, and in particular to version the data objects or content of an edition independently from the environment in which it is displayed. The article concludes with a case study of the development of a versioning policy for the *Piers Plowman Electronic Archive*, and includes an appendix surveying how a selection of digital editions handle the problem of recording and communicating changes.

Introduction

Digital editions can change long after publication: errors can be corrected; new materials can be added; the scholarship can be updated. The fact that such changes can occur on an ongoing basis is both one of the great potentials and one of the great terrors of digital scholarly resources. Printed books are comparatively static; while students of bibliography know that changes to books can and do happen in the course of a print run, in most circumstances readers instinctively recognize G. Thomas Tanselle's "central truth . . . that books are not meant to be unique items and are normally printed in runs of what purport to be duplicate copies" [Tanselle 1980, 18].^[1] Moreover, printed volumes are self-identical: a single copy of a book remains the same object and carries the same text unless acted upon by outside forces, like the environment, natural deterioration, or a human hand. Standards for scholarly citation, which solidified around print resources, take advantage of this objectual stability. Referencing a book means identifying its author and title, its edition number (if specified), and the details of its publication (perhaps including the year of its most recent printing or issue). Where the details of a particular copy matter (for instance for incunabula, or where the argument is bibliographic), the writer might go so far as to specify a library or archive and shelfmark. Armed with that information, readers can find and consult an appropriate copy.

1

By contrast, online digital resources can be expanded or corrected long after their initial release^[2] — not necessarily by publishing a new edition that can occupy the shelf beside the previous, nor even through a stop-press correction that will affect volumes printed after it is made, but simply by updating some files on a webserver, with the result that anyone accessing the resource from that point on will see the revised form. Indeed, this mutability is one of the defining promises of digital textuality. Jerome McGann, in his influential essay "The Rationale of Hypertext" (first published online in 1995), contrasts the physical book, which "literally closes its covers on itself" when it is published, with the hypertext archive that "need never be 'complete'" and "will evolve and change over time, it will gather new bodies of material, its organizational substructures will get modified, perhaps quite drastically" [McGann 1995] [McGann 1996, 27, 29]. Less poetically perhaps, but no less significantly, errors can be corrected with relative ease. And anyone who has tried to maintain a digital resource over any duration knows that, quite apart from willful content revisions, changes may not merely be possible but required in order to keep it operational.^[3]

2

While the open-endedness of digital resources, the potential for evolution and infinite expansion, has excited scholars (the creators of large digital editorial projects among them), the inherent changeability of digital materials also poses threats to the scholarly ecosystem. Looking beyond the by now well-known problem of *link rot*, in which online resources linked in references simply disappear from the internet, research on science communication has identified the problem of *context drift*, in which links function but the content on the website has changed since it was referenced [Klein et al. 2014]; a study published in 2016 found that as many as 75% of webpages referenced in scholarly literature in Science, Technology, and Medicine have changed since they were cited [Jones et al. 2016]. Though results would likely be different if examining citations of digital scholarly editions, which are my concern in this article, the issue of *context drift* highlights the problems that digital mutability poses to the scholarly record. Indeed, the Committee on Scholarly Editions of the Modern Language Association (MLA CSE) has identified "the challenge of maintaining the scholarly ability to be referenced in view of the ways that interfaces change over time" as a central issue facing digital scholarly editions [MLA Committee 2016, 7].

3

In this article, I focus on digital scholarly editions, arguing that in order to make sure such editions are citeable and their history is intelligible, their creators and publishers must assign version numbers in tandem with any changes made to edition content. By digital scholarly editions, I refer to any electronic resources that encode textual objects for scholarly study.^[4] While the same considerations might apply to many kinds of digital

4

scholarly resource, I choose to focus on digital editions for a few reasons. For one thing, perhaps more than other areas of digital scholarship in the humanities, digital scholarly editing constitutes a clear community of practice, with a longstanding tradition of editorial theory and a widely (though certainly not universally) shared technical standard in the form of the Guidelines of the Text Encoding Initiative (TEI). For another, the concern with textual histories within scholarly editing and other fields under the umbrella of textual scholarship suggests that editors of all people ought to be particularly attentive to the way textual resources transform in time.

But perhaps most significantly, digital editions occupy a hybrid position in the scholarly ecosystem that makes it especially important to be able to identify and track the changes they go through and the states in which they exist. Digital editions, as I understand them, are simultaneously scholarly publications and data sources. Like all scholarly editions, digital editions are a product of interpretation, scholarly judgment, and the imposition of codes and conventions onto the material being remediated; that is to say, they are works of scholarship, produced according to the research and critical judgment of their creators. Like other scholarly publications, digital editions can be formally vetted through peer review processes.^[5] And in general, digital editions are presented in user interfaces that support the reading and study of the provided text, rather than simply providing encoded files for a user to download.^[6] But editions also provide data on multiple levels. Most basically, they provide texts corresponding to particular documents or works that scholars may reference and cite in publications, treating the edition as a surrogate for the object it edits and using the text it offers as a basis for analysis: that is, as data.^[7] Humans are by no means the only potential consumers of the data embedded in digital editions. Texts and metadata can form raw material for analysis, including computer-aided study and incorporation into large corpora. And the dream of the fully networked digital edition, functionally integrated and cross-referenced by other editions and systems, grows ever more practical with the development of shared infrastructures and standards.^[8] Digital editions, then, are simultaneously publications and data, potentially offering interfaces both to humans and to machines, and it is essential that these multiple consumers be able to understand the evolution of digital editions and precisely reference different states as editions are revised.

Version numbers, I argue, offer a simple and practical method not merely for identifying a state of a resource, but for communicating something of its history and the relationship among its states. Yet defined, citable version numbers still seem to be a rarity in the world of digital editions, and no consensus practice exists in the field regarding how different versions of an electronic textual resource should be identified, or what it is that version numbers should communicate.^[9] Although textual scholarship has created sophisticated frameworks for understanding revision and the evolution of texts, I suggest that software developers have much to teach editors about versioning living resources in ongoing development and publication. This essay argues that a new version number should be attached each time an edition is updated, that version numbers should communicate something meaningful about the scope of changes to the resource, and that the encoded informational content of an edition should be versioned separately from the interfaces through which users access that information. After outlining considerations involved in assigning version numbers, I conclude with a case study of the development of a versioning policy for the *Piers Plowman Electronic Archive*, a longstanding scholarly resource that has published editions of multiple texts in evolving formats.

Approaches to Change

The fundamental changeability of digital resources, including digital editions, poses challenges to longstanding scholarly paradigms of authority and completeness. Kathleen Fitzpatrick has suggested that the capacity — indeed, in some circumstances, the necessity — for digital writing to change over time might suggest a fundamental change in our understanding of scholarly writing, from product to process [Fitzpatrick 2011, 66–72]. But new paradigms only slowly beget the practices needed to support them. Paul Fyfe has argued that we have not sufficiently theorized how digital scholarship deals with the problem of error [Fyfe 2012]. Correcting error is relatively simple, but scholarly practices surrounding correction lag behind.

For digital editions in particular, change is a double-edged sword. Although few editors would now claim to be producing “definitive editions,” the goal of any editor is presumably to produce an accurate text representing the document or work being edited according to that editor’s theory of the object of study.^[10] Thus, the ability to incorporate corrections and continuously present the most accurate possible text in one sense enhances the reliability and scholarly value of digital editions by contrast to print, where errors discovered after publication can be corrected only in later printings or by issuing errata. But that same flexibility underscores the need to be able to clearly identify particular states of the resource. Consider a scholar who bases an argument on a particular reading taken from a diplomatic text presented in a digital edition. The editors later discover that they have made an error in their transcription and update the edition. Without a way to identify the specific state of the resource when it was cited, the error may appear to have been the scholar’s, and the scholarly record is muddled. Similarly, archivists seeking to preserve a digital edition can more effectively capture its history if the resource clearly signals when changes occur. And computer systems that ingest and process data from digital editions (for instance aggregating texts from multiple publications, or analyzing the text of an edition and recording statistical information in a database) have the same needs as human researchers: to know in what form they have accessed a resource and when changes have occurred. Citation styles, clerical practices, and technical measures have all attempted to offer solutions to the problem of digital change, but I argue that explicit versioning of resources can more effectively meet the needs of digital reference.

When citation guides were first faced with the problem of the mutability of digital resources, some suggested that researchers citing online publications should include in their citations the date on which they accessed the material [Gibaldi 1995, §4.9.1] [Turabian et al. 1996, §8.141] [Gibaldi 1998, §6.9.1] [Publication Manual 2001, 4.16.71ff].^[11] Access dates recognize that online resources evolve, but they are concerned with a researcher’s activity (visiting a website) rather than with the resource itself. Unless a resource happens to have been archived on that particular day, a date of access does not point to a particular form of the material (and there is, of course, no guarantee that the resource did not change later on the day it was cited). *The Chicago Manual of Style* accordingly suggests that “access dates in online citations are of limited value” and

does not recommend including them in citations [Chicago Manual 2017, §14.12]. And for computer systems interacting with a resource, recording the date of last access does nothing to determine whether it has changed since that last access.

Another way of dealing with the problem of mutating resources — the method most endorsed by the TEI, and the most widely used way in which digital editing projects appear to deal with textual change (see Appendix) — is manually creating a log of revisions. Attaching revision metadata to files allow records of revisions to be closely associated with the files themselves, though this information does not make it possible to reference particular states of the text. The TEI Guidelines provide the XML element `<revisionDesc>` in the header of each file to record narrative explanations of changes and the reasons and agents behind them in individual `<change>` elements associated with each revision [TEI Consortium 2019, §2.6]. Because change logs are simply written records of modifications, they are not tied to the TEI, or to any particular metadata format. The *Walt Whitman Archive's* [Folsom and Price n.d.], for example, maintains a public change log in the form of a blog that provides clear descriptions of modifications to the *Archive*, from corrections of typos to pervasive metadata updates [Walt Whitman Archive Changelog 2019].^[12] Individual XML files also carry the TEI `<revisionDesc>` element. *The Whitman Archive's* approach models thoroughness and transparency in disclosing ongoing modifications to a digital resource. But the *Archive* largely obscures these revision histories from users of the site. The change blog is hosted at a different web domain from the *Archive* itself, and the revision lists embedded in file metadata are not displayed in the reading interface provided on its website — probably the context in which most users will encounter the texts. In contrast, the *William Blake Archive* [Eaves et al. 2017] extracts this revision history and presents it in a human-readable format in an Electronic Edition Information section associated with each object in its collection. This section of the display makes the file history more directly available to readers conducting research within the *Archive* and conceivably allows readers to cite the date of the last revision, but still does not supply a specific identifier pointing unambiguously to a particular state of the file. The many editorial projects that use change logs store and expose that information in a wide variety of ways, but share an interest in recording what kinds of changes were made, when, and (often) by whom — without necessarily offering a way to reference a state of the resource resulting from a particular set of changes.

Nor do change logs offer any way to get back to prior versions of a resource; a user can understand what has changed, but not access an earlier form. The increasing embrace of revision control systems (RCSs) such as Git in the digital humanities has suggested the possibility of automated, systematized methods for tracking revision history and providing access to specific states of a project or file.^[13] Elena Pierazzo proposes that RCSs should be embedded in digital edition software, exposing the evolution of an edition and providing access to previous states [Pierazzo 2015, 185–186].^[14] Wiki-based editions, such as *A Social Edition of the Devonshire MS* [Siemens et al. n.d.], are one existing model enacting Pierazzo's hope for editions with built-in RCSs. Christian Wittern goes even further, suggesting that distributed RCSs such as Git might furnish a new ecosystem for scholarly publishing of digital editions, allowing the maintenance of fine-grained revision histories as well as the coexistence of multiple revisions of a single file carried out by different scholars [Wittern 2013, §4].

RCSs make file history accessible, but do not necessarily identify or make intelligible meaningful developmental stages. While different RCSs provide different features, broadly, they operate by storing the content of each file as well as a precise record of each change made to any file. As a result, all changes are reversible, and it is possible to retrieve any previous state of a file as it was stored in the RCS repository, as well as any previous state of the repository as a whole. In order to facilitate retrieving earlier states, RCSs do (unlike change logs) provide unambiguous identifiers for a particular state of a file. In Git, for instance, each commit has an associated hash: a cryptographically generated key that can be used to identify and retrieve a particular state of the repository. A particular version of a file, or of the whole project, can thus be identified through an associated hash. However, they are not necessarily meaningful to human users. Git hashes, produced using the SHA-1 algorithm, take the form of forty-digit hexadecimal numbers (usually cited only by their first few digits). The hashes of successive commits bear no visible relationship to each other; indeed, given two hashes but no access to the repository containing the data, it is not possible to determine which represents the more recent state of the data. Other RCSs use different mechanisms, some of which are more straightforwardly numeric, but identifiers within RCSs are inevitably tied to the details of the system and may not correspond to human editors' understanding of their processes. They cease to identify states of a file or resource if that file is archived elsewhere, or even if the project migrates to a new RCS.^[15]

Revision control is an important tool for data management in the digital humanities. But explicit, deliberate versioning of data should go beyond recording revision history or providing an arbitrary identifier for a particular state of a file (bound to a specific RCS). Versioning should communicate information that helps both humans and computers understand how that version relates to others and the context in which users should approach it. Assigning version numbers to digital editions would permit humans and computer systems not only to refer to a particular state of the edition, but to understand the relationship between any two copies. Adopting clear versioning practices aids both the preservation and the reuse of data, and the producers of digital editions can benefit from practices developed both in the fields of textual scholarship and software development in producing useful version numbers for digital editions.

The problem of versioning data is by no means unique to digital editions; it is a pressing issue of research data management and publication across disciplines. The W3C recommendation on “Data on the Web Best Practices” highlights the importance of versioning data and specifically indicates the value of standardized, meaningful version numbers that not only identify versions, but suggest how they differ [Lóscio et al. 2017, §8.6 and Best Practice 7]. But despite increasing recognition of the importance of clearly versioning research data, standard practices around versioning have yet to cohere in the research data community; a guide to data versioning from the Australian National Data Service is replete with language like “no agreed standard or recommendation” and “no one way” [Australian National Data Service n.d.]. Still, emerging data infrastructures support a move toward more transparent and explicit versioning. For instance, the research data repository Zenodo introduced support for versioned Digital Object Identifiers (DOIs) in 2017, allowing depositors to update their data and permitting researchers to cite both specific versions and a whole concept independent of version [Nielsen 2017].^[16] Particularly within the Open Science movement, the growing emphasis on data publication has translated into attention on data versioning.

However, the textual digital humanities, and digital editing in particular, have been slower to adopt versioning practices. Editors' awareness of the problem of textual variance should make them more attuned to the need to track and publicize the evolution of their own editions. The practices of textual criticism point to the value of developing versioning protocols for digital editions.

15

Textual Versioning

When textual scholars use the term *version*, they mean something different from (though related to) the way the term is used in software development. Because my argument that digital editions need versioning policies lies at the intersection of these fields, it will be useful to survey the ways in which the two fields think about versioning. Literary scholars and textual critics might speak of the Quarto and First Folio versions of Shakespeare's *King Lear*, or the A, B, and C versions of the fourteenth-century alliterative poem *Piers Plowman* — or to distinct draft versions produced during the course of Thoreau's revision of the single manuscript of *Walden*. Software developers (or users), on the other hand, might refer to version 5.2.1 of the Linux kernel, or to Apple's iOS 12.4 (eliding the word "version" entirely). Do these concepts have anything to do with each other? Though their orientation is different — textual scholarship is focused on historical analysis, software development on ongoing maintenance, publication, and support — both fields share a common concern with making it easier to understand variation and evolution, a concern likewise relevant to the problem of changes in digital editions.

16

"Version", as used by textual critics and editors, generally denotes a distinct state of a work or a document that has transformed in time. Literary works exist in different versions because of alterations during the course of their composition and transmission — alterations by the author or by someone else, willed or unwilling. So, a campaign of authorial revision of a work would produce a new version of that work, as might the copying of a medieval manuscript in which a scribe introduced changes (even unintentional ones), or the publication of an expurgated edition long after an author's death. These versions have independent value as forms in which creators conceived and audiences encountered the work. Donald H. Reiman in 1987 argued for what he called "versioning", as a counterpoint to editing: rather than producing complicated, expensive critical editions, he suggests, it may be more productive to publish accessible texts of major forms in which a work existed, such as important editions and authorial manuscripts, allowing readers to compare the texts themselves [Reiman 1987]. (The profusion of digital documentary editions suggests that Reiman's dream is increasingly being realized.)^[17]

17

Both genetic critics and those concerned with the "sociology of the text" have emphasized the coherence and vitality of individual versions of developing works, pointing to the inadequacy of the notion of final authorial intention and calling for editorial and critical engagement with versions as coherent units.^[18] Hans Zeller argued that individual variants in witnesses to a work cannot be considered in isolation, as had been common under the principles of eclectic editing; rather, we must recognize "the relationship of its elements to one another and to the whole, and therefore to what constitutes a text as a text, to what makes it into a particular version" [Zeller 1975, 237]. Peter L. Shillingsburg identifies the concept of version as "a means of classifying copies of a Work according to one or more concepts that help account for the variant texts or variant formats that characterize them" [Shillingsburg 1991, 50]. A version is thus a concept, not a thing; it is distinct from any physical embodiment (which might not represent it reliably), and versions come into being through the act of reading, as readers create them to organize textual variants [Shillingsburg 1991, 51, 73]. John Bryant, articulating his concept of the "fluid text" defined by the flow among different versions, echoes the notion of versions as "critical constructs" but also emphasizes their relationality: all versions exist in relation to other versions; they come into being through revision (which may or may not be intended); they are "pulsings of . . . collective energy" that can involve both authors and the editorial and cultural forces surrounding and following them; they have their own conceptions of the work and speak to their own readerships [Bryant 2002, 88–90]. While these and other theories of the concept of version differ on points such as the precise degree, nature, and agency of the changes that can produce a new version, they share a sense that versions are distinct and alive, and their coexistence is part of what constitutes a work.

18

These sophisticated frameworks for textual change may seem far from the problems of labeling changes as a digital edition is revised, and from the straightforward numerical approaches that I will draw from software development. But textual-critical accounts of versioning remind us that readers (and, we might add for our purposes, machines) encounter individual texts as coherent units, and these discrete forms have existence and meaning independent of the work as a whole. The kind of versioning this article focuses on is not teasing out key moments in the life of a work that is the object of study, but identifying moments of change in the evolving life of the published edition. In other words, echoing Hans Walter Gabler's understanding of the contents of an edition, this article is concerned with versioning the editor's text and the editorial discourses attached to it [Gabler 2010, 45]. An edition might present one or more versions of a work or of a document (indeed, the ability to present more versions in more dynamic forms has long been heralded as one of the most exciting potentials of digital editions), but what I address in this article is the need to version that edition as an edition, to keep track of the changes that occur within the edition itself.^[19] Versioning, as I use the term, means assigning version identifiers to public materials as they develop; it is a publication practice rather than a critical practice.

19

That leaves the practical problem (unresolved by textual-critical theories of versions, which focus on more complicated analyses) of how to communicate the state of the edition to its users, whether humans or software programs. Existing publishing practices a not great help. Print publication simply has not developed conventions for describing ongoing revisions. Minor errors discovered after printing might be dealt with by issuing a list of errata; a more thorough revision might occasion the publication of a new edition. This publishing logic features in the closest the TEI Guidelines come to addressing the versioning of texts. The `<editionStmt>` section of the TEI header groups together information about an "edition" of a TEI-encoded text [TEI Consortium 2019, §2.2.2]. The Guidelines link the intellectual foundations of the concept of edition to the idea of a "master copy", while simultaneously noting that the concept does not really apply to electronic texts. Nevertheless, the primacy of the print concept of edition leads the Guidelines to distinguish between "substantive changes" (such as the encoding of new information throughout the file) and "minor changes . . . which do not amount to a new edition" (such as error corrections or conversation between encodings) — a distinction that

20

the Guidelines themselves acknowledge to be somewhat arbitrary and subjective. These “minor changes” can be recorded in `<revisionDesc>`, but there is no mechanism for labeling them. Confusing the issue still more, the Guidelines treat edition as synonymous with version, level, and release, while using the terms revision and update for minor changes below the level of edition. Finally, the Guidelines offer two rather different ways of recording version information in the same element. The edition (or version) can be recorded either descriptively, with a phrase like “new edition” as the content of the `<edition>` element, or with a “formal identification (such as a version number) for the edition” in the `@n` attribute. The Guidelines introduce a concept broadly similar to the print concept of edition, but one that lacks the technical underpinnings (the setting of type) that gave the concept its meaning in print, and that lacks the expressive power for dealing with digital textuality in a comprehensive way.

Technical Versioning

The shortcomings of the TEI’s print-inspired model suggest that we might look elsewhere for models to describe changes in computer-encoded data files with sufficient granularity. The field of software development has, over a period of decades, developed software version numbers as a system of practice for tracking the development of complex, digital objects — pieces of software — as they are published and revised. Software version numbers also situate the objects they describe in their developmental histories, but from the inside: rather than analytically describing objects after the fact, they are assigned during the development and release process to track ongoing work. Software version numbers facilitate many kinds of reference: they track changes to a piece of software, help users know when updates are available, facilitate technical support by unambiguously identifying a particular state of that software with all its particularities, and promote interoperability by allowing computer programs to determine whether they are compatible. Despite its straightforwardness, software versioning is a rich signifying practice, and it offers a model that suggests practical solutions for editors of digital editions.

21

Version numbers, at their most basic, delineate stages in the development of an object — for instance, a piece of software — by quantifying them and assigning ordered numbers to the object. It would be possible in principle to use a single whole number, which increases with every change. However, this approach, which fails to distinguish the scope of the changes that have been made, is insufficient for dealing with complex software objects. It is instead common practice to subdivide the version number into parts according to the scale of the difference from what has come before. The most common approach is to segment the version number, using a period to divide the parts. Version numbers with either two or three segments are common. A piece of software with version 2.7.4 would thus signify major version 2, minor version 7, revision or patch 4.^[20] (The meanings of the first two numbers are typically major and minor version; what, exactly, later numbers communicate is less consistent, though they often indicate small revisions intended to fix errors without adding features or altering behavior.)

22

The meanings of these sequences are not fixed; different software creators are free to construct their version numbers in different ways, and there are no universal criteria for distinguishing major and minor releases — although some recent efforts, which I will discuss, have attempted to make version numbers more systematically intelligible. But broadly, major version releases are likely to introduce significant changes to a product: for instance, a new user interface, a large set of new capabilities, or technical changes that make files produced with the new version incompatible with previous versions. Minor versions might introduce features that do not substantially alter the nature of the product, or correct problems that have been discovered. Smaller releases, like patches, are likely to fix individual errors.

23

This way of conceptualizing versions is at heart hierarchical, with each level in effect “containing” those below it. In general, bumping the version number at any level resets all the levels below it to zero, so that, for instance, the major release that follows 2.4.7 is given version number 3.0.0. Conceptually, the life of a major version consists of all the releases under that major version number, not just the original point zero release. This hierarchy roughly parallels the way the edition–impression–issue–state model subdivides the bibliographic object (see Bowers (2005), 37–42, 406–411; Tanselle (1975)). An edition, in bibliographical terms, is created whenever a given text is typeset; setting new type constitutes a fundamental change in the essence of the object even if the text remains unchanged. A new edition is a kind of new major version, an object that on some level shares identity with what came before but also represents a significant break. Other categories are grouped beneath this, expressing different levels of identity change with an edition subdivided into impressions (the copies printed at once) and impressions divided into issues (copies intended as a unit of sale) (see Tanselle (1975), 28n14). Sheets of books even get “patches”, changes correcting individual errors; in his attempts to distinguish issue from state, Fredson Bowers suggests that minor textual corrections, along with small supplements, produce only new states and not new issues because they are simply “delayed attempts to construct an “ideal copy”, much as software patches do not seek to extend functionality or change intended behavior but merely make the software conform to existing expectations [Bowers 2005, 67].

24

The point, of course, is not that software versioning and bibliographic description map the same procedures to different media. Each practice is informed by different practical needs, disciplinary contexts, and underlying technologies. Rather, I wish to point to a broad correspondence in approach between the two procedures, even though they bear different relationships to their subject matter: both organize intellectual objects hierarchically, categorizing and subdividing around questions of essential identity and of imagined ideal state.

25

But bibliographic classification, as an analytical practice, is rooted in the evidence of specific changes. Software versioning, by contrast, has been accused of being arbitrary and inconsistent — and at times of being driven by market forces rather than technological logic. A few efforts to make versioning practices more consistently meaningful help clarify what version numbers can actually assert about an object.

26

Calendar Versioning

One approach to versioning software, which has been called Calendar Versioning, highlights the temporality of releases [Hashemi 2019].^[21] This approach recognizes that knowing when a software object was released may be the most important way to identify and evaluate it. Microsoft has offered the most widely visible version of this practice, with releases like Windows 95, 98, and 2000. (It is worth noting, however, that these are

27

merely public release names and the software actually carries a different version number distinct from the release name.) But a variety of other software uses Calendar Versioning in less dramatic ways: the Ubuntu Linux distribution, for instance, offers what look like fairly traditional version numbers, but the first segment of the version numbers is the last two digits of the current year, followed by the month, so that as of the time of writing, the most recent version (released in April 2019) is 19.04. This approach has appealed to at least one digital editor; the texts edited by Jeffrey C. Witt from Peter Plaoul's commentary on Peter Lombard's Sentences carry version numbers that employ a form of Calendar Versioning, as detailed in the Appendix.

In emphasizing date as what identifies an object, calendar versioning resembles scholarly citation practices, which emphasize publication dates, and sometimes access dates — although calendrical version numbers clearly and uniquely identify particular resource states, as access dates do not. Calendar Versioning privileges temporal sequence above all else; it suggests that when an object was produced is the most salient information for assessing it. It also establishes a sequence of versions, chiefly by relating them in time. Thus, Calendar Versioning is effective for allowing users to assess the age of a particular resource, to understand which versions were produced earlier and later, and to determine whether a more recent version is available. But it does not indicate not scope: it is impossible to tell from version numbers alone whether two versions are differentiated by the correction of a minor error or by a significant overhaul.

28

Semantic Versioning

Even generic versioning practices have tended to make the degree of difference among versions greater than in Calendar Versioning, differentiating major and minor versions according to the relative degree of change. However, these approaches have appeared inconsistent to some critics: different developers or companies make different decisions regarding what constitutes minor and major versions, and these decisions are sometimes driven by market forces as a new major version might generate excitement or drive customers to upgrade. The Semantic Versioning specification, created by Tom Preston-Werner, is an attempt to specify rigorously and technically what version numbers (or, more precisely, what changes in version numbers) actually mean [Preston-Werner n.d.].^[22] I will dwell slightly longer on Semantic Versioning, because it has provoked a debate that exposes a fundamental question not merely of how versions should be identified but what versioning is for — a debate that helps expose for the creators of digital editions the role that versioning practices might play in communicating with the public and interfacing with larger systems.

29

Semantic versioning is based on the traditional [major].[minor].[patch] format, but attempts to codify something largely implicit but inconsistently practiced in community practices for giving version numbers to software: that the different portions of a version number reflect different kinds of change. Semantic versioning is most concerned with libraries and packages (that is, pieces of software designed to be used by other pieces of software), and specifically with what are called their APIs (Applications Programming Interfaces): the formal methods through which other programs interact with the package. (It is worth noting, however, that the Semantic Versioning specification is itself semantically versioned; the application of these principles is not restricted to packages or libraries.)

30

Semantic Versioning is primarily concerned with whether changes to a package break backwards compatibility. That is, have you changed the way your API works so that the same command, issued to the new version, will produce different results? The central principle is that any breaking change to the API (that is, one that will cause the same command to have different results) is a new major version. A release that adds new functions while maintaining backwards compatibility is a new minor version, while a patch version is one that simply fixes bugs, provided the fix does not break backwards compatibility. Semantic versioning is designed especially for use with package managers, programs that can automate procuring and updating the packages needed to build or run a piece of software.

31

Semantic Versioning sits especially uneasily at the intersection of intellectual and mechanistic understandings of versioning. Jeremy Ashkenas, an influential JavaScript developer and vocal critic of Semantic Versioning, argues that the system “prioritize[s] a mechanistic understanding of a codebase over a human one. . . . It's alright for robots, but bad for us” [Ashkenas 2015].^[23] Ashkenas suggests that, in an environment where other developers may write source code relying on a project's bugs, the definition of a “breaking” change is subjective — a point others contest. Perhaps most significantly for Ashkenas and other detractors, small function changes might under Semantic Versioning require an increase in the major version number (say, from 2.3.1 to 3.0.0) — a change that implies a major rethinking of the software that may not, in fact, exist (and can cause version numbers to balloon). Ashkenas agitates in favor of what others disparagingly call “Sentimental Versioning” and he playfully labels “Romantic Versioning”: a system under which a developer's understanding of the magnitude of the change and the relationship between versions defines the version number.^[24]

32

The crux of the debate around Ashkenas's rejection of Semantic Versioning, which riled a community of developers whose projects were affected by an update that Ashkenas declined to label a new major version, is whether version numbers are intended for human or machine consumption. Software processes that decide whether it is safe to update a given library do not care what a developer's sense of the change is; humans, on the other hand, may be misled by seeing a major release that actually consists of a conceptually minor change.

33

Why should scholars at the intersection of physical book study and digital scholarship be concerned with a four-year-old squabble among software developers, much of which involved how developer practices integrate with automated systems? The Semantic Versioning debate is particularly interesting for digital textuality because it draws attention to the different kinds of weight that version information can carry, and the different systems into which it integrates. Software developer Niels Roesen Abildgaard has attempted to nuance the Semantic Versioning debate by suggesting that software exists on a continuum between interfaces directly with users and interfaces exclusively with other software; user-facing software, like games or (to a lesser extent) desktop applications, is most suitable for Romantic Versioning, since human understanding is paramount, while software libraries would benefit from Semantic Versioning because relatively few humans will look at them directly, but they will

34

often be included in other software systems [Abilgaard 2015]. The Semantic/Romantic debate draws our attention to the fact that version numbers provide an interface for understanding software changes, and that this interface is conditioned by purpose and audience: a key insight for considering how users of digital editions might interact with version numbers and what information they can convey.

The focus on versioning as a communicative interface, designed to work in a system with a clear audience to satisfy a defined purpose, helps us understand the complexity of digital editions as objects to be versioned. Digital editions operate within multiple systems at once. They are typically created first and foremost as objects for *reading*, to be studied closely by individuals. They are also sources of data, furnishing both character data and metadata that can be manipulated and analyzed in a variety of ways. And they are objects of citation, which must be unambiguously referenced in scholarly environments. McDonough et al. point out that people using and analyzing digital objects for different purposes may have profoundly different (though interrelated) needs in terms of how they are categorized [McDonough et al. 2010, §18].

Moreover, digital editions are complex, layered objects. At base, they consist of one or more transcriptions or constructed texts, which may have been collated or further analyzed to produce altered texts. In most cases, these texts have been encoded using a markup language to identify features, define structure, and incorporate metadata; at the level of information, they are accompanied by products of scholarly analysis, such as a critical apparatus and various annotations. And in most cases, they are accessed through a software interface for reading, which may well be unique to the edition or project in question, and perhaps through APIs that provide data upon request. Even if the content of the edition began life encapsulated in a manageable format like a single XML file, the reading interface will encompass a multitude of files and technologies, like CSS and JavaScript files executed on a user's computer and other processes, such as XSLT transformations, that may occur on a server entirely out of a user's sight (so that the user may not even directly receive the underlying data files without requesting them). Any APIs the edition provides will operate similarly, extracting and transforming data to answer the requests it receives.

Describing Electronic Literature

Given the complexity of digital editions as textual objects, one place we might turn for more robust ways to describe them as temporal, bibliographic objects is work done by electronic literature scholars in classifying and categorizing their materials. (Digital editions are indeed a form of electronic literature, albeit one that has not attracted much study outside the field of editorial theory.) Matthew Kirschenbaum in a 2002 article postulated a set of terms for describing first-generation electronic objects inspired by Bowers's classic bibliographical typology [Kirschenbaum 2002]. *Layer*, *version*, and *release* refer to the whole software object — another hierarchy. *Layer* refers to a whole integrated environment of software and data; adding a brand new software interface, for example, might constitute a new layer. *Version* is somewhat subordinate to layer and describes the life sequence of the software; a new layer creates a new major version, while refining an existing layer creates a new minor version. *Release* seems to be primarily a matter of distribution channel: releases are “computationally compatible . . . but . . . not functionally integrated”, and Kirschenbaum's example is of a work released both online and on CD-ROM (presumably with the same underlying software) [Kirschenbaum 2002, 48].

Within the total software object so described are individual objects — individual digital entities. Kirschenbaum offers a file as an example of an object, but it is worth noting that Kirschenbaum's objects are independent of the data format in which they are stored. These are described by *states*: “the computational composition of an object in some particular data format.” For example, separate PNG and JPEG files representing the same image are different states of the same underlying object. Instance exists at the interplay between state and the software environment in which it operates: an image displayed in a particular program, which might (intentionally or inadvertently) render it differently from other programs. And finally, there is *copy*, a single instance of a state of an object, for example, the copy of an image that a web browser downloads and stores on a user's computer (as distinct from the copy on the server).

I rehearse this categorization at length because it represents a particularly thorough and robust attempt to think through the distinctive properties of electronic objects, and it, too, points to some of the properties we must consider when evaluating digital editions. Kirschenbaum's seven-part system is certainly too detailed and cumbersome to be used as a versioning system in itself — though a refined version, adapted for the era of networked publishing, might ultimately prove valuable when scholars of future decades write bibliographic accounts of digital editions. But his approach might suggest what sorts of features versioning needs to account for.

From the perspective of digital editions, the central insight of Kirschenbaum's proposal is his distinction between the whole software environment and the individual components that compose it. An electronic text, consisting of both data and a technical environment in which that data is remediated for a reader, cannot usefully be described in total; media objects simultaneously precede their instantiation in a particular technical environment and become entangled in the systems that display them. Kirschenbaum applied this schema to works of electronic literature; those discussed in his account appear to have evolved in relatively well-defined, separable releases that can be thought of as an issue of all parts at once. His descriptive vocabulary seems to reflect this tendency: terms addressing the whole software environment are concerned with evolution, while those concerned with individual objects are concerned with instantiation.

This particular division would work well for describing digital editions of the CD-ROM era, where the production of physical copies created a distinct issue of the whole, including both data and display software. But versioning all the parts together appears less appropriate for the “continuous publishing” practices of the web era, where individual components (and most importantly individual documents) may be updated independently, not to mention the prevalence of digital editions in large archives containing many documents, and even in semi-distributed systems like Jeffrey Witt's Scholastic Commentary and Text Archive (SCTA), which promises to aggregate related, interoperable editions [Witt n.d.] [Witt 2018]. If Peter Boot and Joris van Zundert are correct that distributed, networked systems combining many data sources and services are the future of digital editions (“the digital edition 2.0”, as they call it), versioning all the components of an edition as a single unit may well

become completely impossible [Boot and Zundert 2011].

Versioning Object and Environment

Accordingly, rather than versioning whole systems, we should offer separate treatment of objects and environments. *Objects*, here, are the edition content: the texts or other resources being presented online, as represented in the edition (*not* the physical objects being edited). *Environment* describes the whole system within which these objects are rendered and consumed: a web of server-side and client-side electronic processes that work in tandem with a user's local computer environment to display an edition, or that provide data to other systems upon request.^[25] Put another way: objects are the underlying data, the textual and editorial content that editors create and incorporate into the edition, regardless of the specifics (technical or visual) of its realization. Environment encompasses the interfaces through which that data is made available to users (the visual layout of an edition on the screen, APIs that permit machine-driven access to edition data), as well as the software environments that enable these forms of access.^[26]

42

In arguing for the separation of object from interface, I do not mean to imply that interfaces are “mere” technical contributions, separate from the intellectual work of editing.^[27] Nor am I suggesting that a reader or user of an edition can experience content in some pure way, uninflected by the way it is presented. The layout (interface) even of traditional print editions constitutes an argument about the material and its character [Eggert 2013]. The vaunted flexibility of digital editions means an edition may contain and present its material in multiple ways, indeed, through quite different interfaces, yet these interfaces will inescapably condition the material and make arguments about understanding it [Andrews and Zundert 2018]. Certainly, the form in which a text is encountered conditions understandings of that text, and citing the environment in which it is encountered will be necessary both to understanding conclusions drawn from the edition and to recognizing intellectual contributions to it.^[28]

43

Despite the entanglement of content and presentation for scholarly understanding, versioning objects separately from their environments has both intellectual and practical benefits. One influential idea in software design, and key principle of the modern web, is the separation of form and content: the principle that documents should be encoded according to their underlying structural logic, without intermixing instructions regarding how that content is to be displayed.^[29] The separation of form and content has understandably been drawn into question by scholars gesturing toward the outpouring of work on the materiality of text.^[30] Nevertheless, this distinction operates at a technical level in many digital editions, and offers a model by which digital editions may be implemented and preserved. The TEI guidelines, perhaps the most common standard for textual encoding of digital editions (see Franzini et al. (2019), 16), endorse and support the separation of form and content.^[31] C. M. Sperberg-McQueen has gone so far as to suggest that it is a best practice for digital editions to provide multiple interfaces, not just to support multiple ways of interacting with the text but also to force editors to make sure they are not basing their encoding on desired display rather than the logic of the content [Sperberg-McQueen 2009, 35–36]. While others might argue for tighter control over the presentation of an edition as an editorial responsibility,^[32] the edition content and presentation are still technically and intellectually separable even when they are thought of as forming a single intellectual unit. Put another way: Sperberg-McQueen distinguishes among the infinite array of facts concerning a particular text, the selection of facts that are contained as information within a particular edition, and the presentation of those facts, for instance through arrangement on the screen. The selection of facts — the total information content available in an edition, whether exposed in a particular form or not — exists, as encoded data, apart from the mechanisms that present those facts, even where the selection of facts and the development of the user interface have informed each other and where they are intended to go together [Sperberg-McQueen 2009, 31].^[33] Unless a creator goes to extreme lengths, against all norms of software design, to create a boutique piece of software in which data and display are fully entangled, it is likely that any digital edition (regardless of the standards or ad-hoc principles followed) will contain data objects that can be meaningfully versioned apart from their display systems.

44

Moreover, at a practical level, objects and environments are likely to evolve separately, both before and after an edition is published. An editor who learns of an error in a reading can correct it by making a change in a data file; in many online publishing environments, no further action will be required for the correction to appear in the edition.^[34] Similarly, the developers and maintainers of digital editions can often make changes from tweaking the text styling to rearranging the graphical user interface to adding major new features for textual analysis without altering the data files. Versioning data objects can also aid preservation. Because the underlying data in most edition objects is at heart textual, data files can be relatively easily archived in repositories designed for storing texts, like the Oxford Text Archive [Oxford Text Archive n.d.] and TextGrid [TextGrid Consortium 2006-2014]. Depositing an edition's data is not the same as preserving the edition, and work on the preservation of the interfaces should continue (informed by work in the field of software preservation), but such deposits can help allow the labor represented by an edition to live on as data even if its software becomes inaccessible.^[35]

45

As publishing practices evolve, being able to refer to objects separate from their environments may increasingly become a practical necessity. Boot and van Zundert's vision of a networked, distributed “digital edition 2.0” involves bringing data together with services offered by different providers, and they explicitly argue that editions should not provide their own “advanced services” [Boot and Zundert 2011, 144]. Users of digital editions may already be prepared to work with data apart from interfaces; in a recent survey about digital editions, a majority of respondents rated the ability to download and reuse data from editions “very important” [Franzini et al. 2019, 15]. Even without such a shared infrastructure, thinking separately about object and interface helps prepare us for the future agitated for by Peter Robinson, where digital editors abandon the practice of providing their own interfaces and leave textual display to others [Robinson 2013]. Increasingly, those working in the field of digital editing recognize the value of publication frameworks and software packages that allow editors to present their work without having to develop entirely new software.^[36] Alternative environments need not be software systems; as discussed below, the *Piers Plowman Electronic Archive* has begun publishing printed volumes produced from its XML data, providing paper-based access to the edition. Versioning objects and environment

46

separately means that our versioning practices can recognize the intellectual identity between an encoded document (for example, an XML file in the TEI vocabulary) and its rendering (for example, its rendering as an HTML page as a result of an XSLT transformation). They remain the same object, even if the mediating layers change.^[37]

Although the remainder of this article focuses on versioning data objects, versioning software environments (including the web platforms that display digital editions) is also important for the future health of the digital scholarly ecosystem, and should be an area of further work for the field. A FORCE11 working group has emphasized the importance of scholars citing the software they use in their research, for reasons of credit, provenance, and reproducibility, and has indicated that one goal of software citation should be to identify and facilitate access to a specific version of the software [Smith et al. 2016]. Although the primary focus of software citation movements has been software executed locally by researchers, such principles might furnish a starting place for citation of web-based environments in which the contents of digital editions are accessed. Publishers of digital editions might facilitate such citations by assigning their online software platforms specific version numbers, incremented with every update, even when the web interface is specific to a single project. Researchers citing a digital edition might then cite both the data underlying the edition and the platform in which they accessed the edition. Publishers might also consider making the source code of online platforms available under open source licenses, potentially enabling future researchers to recreate an earlier version of an online platform that has since been updated or discontinued. Of course, such steps are at best partial. The ways in which a user experiences the data mediated by an online edition platform depend not merely on website code, but on underlying elements of the web architecture (such as the specific versions of software running on the web server) and on features of a user's own computer, such as operating system, web browser, and specific settings. Research into the preservation and curation of software as a part of the scholarly record is ongoing, and as the field of digital editing and publication continues to mature, it will need to become involved in these broader conversations.^[38]

But there is lower-hanging fruit for editors and publishers of scholarly editions, who have yet to develop standards for the comparatively straightforward versioning of edition contents, standards that would benefit the field of scholarly editing. Versioning the contents of digital editions would represent a significant step forward for citeability and preservation of the scholarly record even while difficult issues regarding software environments await future work. We can, and should, version the data objects that form the information content of our digital editions, starting now.

Developing Versioning Protocols for *Piers Plowman Electronic Archive* Data Objects

I will turn now to a case study based on my work in creating a formal versioning policy for the *Piers Plowman Electronic Archive* (PPEA) [Duggan et al. 2019], an open-access online resource that aims to document the complete medieval and early modern textual tradition of the Middle English alliterative poem *Piers Plowman* through TEI-encoded documentary editions of individual witnesses and critical editions of archetypal texts. This long-running project, which began in 1987, demonstrates both the need for and the challenge of clear versioning practices.^[39]

The first seven PPEA editions were published on CD-ROM, from 2000 to 2011, in separate partnerships between the Society for Early English and Norse Electronic Texts (SEENET) and the University of Michigan Press, Boydell and Brewer, and the Medieval Academy of America. The first two CD-ROMs were encoded in SGML presented using the proprietary Multidoc Pro SGML browser; later editions were encoded in XML and published using software that ran within a web browser. In 2014, all texts were made openly available online, in a new web interface created by the Institute for Advanced Technology in the Humanities at the University of Virginia. The new online *Archive* saw the release of previously unpublished editions; older editions were updated to XML conforming to the P4 version of the TEI guidelines. Since 2014, intermittent changes have been made to the appearance and function of the web editions. Forthcoming updates will create additional versions of existing texts: the *Archive* is in the process of updating its texts to TEI P5, and the newly launched PPEA in Print series publishes print volumes derived from electronic texts.^[40]

In addition to changes in medium, file format, and technical infrastructure, the PPEA, like any project of its age and scope, has had to deal with errors in its materials. The web versions of the texts were updated to correct known errors. These changes were not explicitly recognized on the pages for the text. For texts originally published on CD-ROM, the website used to provide Errata lists recording corrections to the CD-ROM texts. However, these lists are no longer maintained given the age of the CD-ROMs, and Errata lists were never created for texts first published online. The corrections made to files spanned a wide range of types and significances, including changes to the format of line numbers (but not the lineation), minor changes to markup unlikely to affect the output on the screen, and the correction of textual errors.

As part of a CLIR Postdoctoral Fellowship in Data Curation for Medieval Studies at the North Carolina State University Libraries, I set out to create standards for assigning version numbers to texts. My primary goals were (1) to allow users of the *Archive* to record and cite unambiguously which version of a text they consulted; (2) to permit previously published versions of texts to be archived and retrieved; (3) to make the history of a given text legible; and (4) to allow users with references to two versions of the same text to have a basic understanding of the relationship between them. From the start, I was concerned only with versioning published resources; while we might use prerelease identifiers to track the evolution of unpublished resources internally, what I sought to define was how we would assign version numbers to editions beginning at the moment of their publication and encompassing all successive published changes.^[41]

A few fundamental decisions guided my work. One early, crucial question was what resource was actually being versioned. First, guided by Kirschenbaum's work, I concluded that edition content and the way that content is displayed cannot be described by the same version numbers. While versioning our display software is a long-term desideratum, my immediate goal was to version editions' informational content. Accordingly, any version numbers we provided would have to refer to the source files for an edition — in this case, the TEI-encoded XML — rather than to its rendered text. The decision to privilege the XML made sense as the XML files can be easily archived, and because it recognizes the markup of an

edition as a significant intellectual product. Versioning the XML files also allows us to link them with any derivatives produced from them: derivatives which can include not just electronic renderings but print volumes.^[42] For instance, editions published in the PPEA in Print series carry a statement on the copyright page declaring the version of the XML files to which the print text corresponds.

Choosing XML files as the objects of versioning has additional consequences. The component files of an edition will be versioned separately. Each full edition consists of, at minimum, separate XML files for the introduction and the edited text. If the versioned objects are XML files, a change to the text does not affect the status of the introduction. Even though the PPEA conceives of each edition as a single coherent publication, and they are peer reviewed as integral wholes, they are made up of separate data sources whose version histories must be managed independently. (This is a more practical approach than creating data packages versioned as a single unit because it allows us to include a file's version number within the file itself without having to modify files that have not otherwise changed.)

One more question concerned what resources must actually be versioned. The PPEA website contains many pages with background information and supplementary resources that are not part of individual editions — some of which, such as site credits, may change frequently. Further, editions include files such as prefaces that are not necessarily advancing the same sort of scholarly claims. At least for the time being, I decided to version only content subject to peer review, meaning the text, apparatus, and introductions of editions.

Establishing some priority of changes that gives a sense of their scope is essential to make version numbers useful. Thus I sought to distinguish changes on the grounds of their scale and significance.^[43] Changes that systematically affected the editorial or markup approaches to a file seemed to constitute a highest level of change. A file's markup might change completely — it might, indeed, be recreated from the ground up in a new format (for instance in HTML rather than XML) — without any differences being visible to users of the edition. However, given the intellectual significance of the way a text is marked up, these two files would be radically different from each other as data. Accordingly, I reasoned, the conversion of a file from one markup language to another (from SGML to XML, or between major versions of an encoding scheme, like the transition from TEI P4 to P5) would constitute a major release (at the highest level of versioning), because even if the intent is to keep the textual content the same, the different affordances of different file formats and encoding schemes mean that the nature of the file has fundamentally changed.^[44] A file modified in this way is incompatible with previous versions in a concrete sense, because changed elements and structure mean that the files can no longer be compared directly to each other by analytical tools that process the underlying XML, and software that worked with earlier versions may not display it successfully. (However, minor changes to how data is stored or expressed, like a switch between minor versions of TEI or a change in character encoding from ISO-8859-1 to UTF-8, maintain the fundamental identity of the file and do not rise to the level of a major release.) Similarly, systematic editorial revisions to a file, I suggested, would constitute a new major release, because they represent a far-reaching editorial reassessment that disrupts intellectual continuity with the existing version. In its relationship to preceding material, a major release is in some ways comparable to a new edition of a print book (marked by a new setting of type), or to a significant version in the text-critical sense.

Since one of the central goals of a digital edition is to present one or more texts, any changes to readings are necessarily significant. I therefore proposed that individual changes to the text that do not rise to the level of systematic revision might constitute a middle level, less high than systematic changes but greater than other forms of change. The concept of a “patch”, a change intended only to correct an error and restore expected behavior, does not apply to an edition, because edition contents may have been used as the grounds for scholarly argument and the change from a mistaken reading to a correct one may thus have great scholarly significance. If changes to text are regarded as the more significant form of local change, then changes to paratext, including editorial content, might be at the lowest level. These three levels of change seemed well suited to the common three-level version number format. I therefore initially proposed that version numbers take the following form: [systematic changes to encoding or editing].[changes to text].[changes to paratext]. I outlined the meanings of these segments as follows:

- The first segment, *systematic changes*, would increase when we make a large number of changes systematically across the text that have a significant effect on its markup or on the how it is edited as a whole.
- The second segment, *changes to text*, would increase when we make any change to our representation of what is on the page. Most obviously this includes alterations of readings, but it also includes highlighting and other features present in the source document.
- The third segment, *changes to paratext*, would increase when we make changes to paratextual content that is not in the source document, such as editorial notes and apparatus.

This proposal sparked discussion with other project leaders. One specific point of debate was the extent to which version numbers should reflect the file history. Following the conventions of software versioning (and of bibliographic classification), I proposed that when any segment of the version number changed, all segments to the right should revert to zero. (So, for instance, version 2.1.3 might be followed by 3.0.0.) Our discussion raised the possibility that this practice hid file history, as after a systematic change it would no longer be clear how many changes to text or paratext had occurred. An alternate proposal was that each segment would increment independently without being reset, so that the number of changes of each type would be permanently visible. That alternative proposal raised its own complications. For one, it deviates from the practices typical of software version numbering, and so would likely prove confusing to users: the practice of zeroing-out later segments is culturally familiar not just from annual demands that we upgrade our phone operating systems, but from its cultural currency in the form of phrases like “web 2.0”. In addition, it creates a false impression of precision, because any number of changes might be bundled into a single update to the file. (For instance, a single update might include three separate alterations to the text and two to the paratext, but the final two segments of the version number would each increase only by one, concealing the actual number of changes.) And there is in any case a hierarchical bibliographical logic to the major version's resetting the clock on other forms of revision: if a major version compares to a new edition, such a significant change establishes a new baseline against which less significant changes can be measured going forward.

Perhaps the most troubling aspect of three-level version numbers from the perspective of the PPEA, however, was the realization that not all files that need to be versioned can have changes at three levels. Files consisting purely of editorial material, such as introductions, do not have distinct textual and paratextual content in the manner of edited texts, and so version numbers would have to express all changes to these files in either the text or paratext segment, with the other segment remaining permanently at zero (or omitted entirely). The idea of versioning different XML files according to different principles, or of having a version number segment that would always remain zero for some files, seemed too unwieldy. And, of course, for readers interacting with and citing primarily editors' comments, it is not necessarily the case that changes to text will be the most significant changes.

59

Accordingly, despite the potential advantages offered by three-part version numbers, we ultimately elected to adopt a simpler, two-part system for version numbers, in the form of [major version].[minor version], where the segments have the following meanings:

60

- The first segment, *major version*, increases when we make a large number of changes systematically across the text that have a significant effect on its markup or on the how it is edited as a whole. Moving from P4 to P5 of the TEI protocols, which requires non-trivial changes in markup across the text, is an example of a change that would increase the number of the major version segment. [45] The significance of any program of changes must be assessed by the resource's maintainers in terms of the needs of the community that will use the resource, as the Semantic Versioning debate suggests.
- The second segment, *minor version*, increases when we make any other change. These include corrections to readings, updates to notes or paratexts, modification of markup, or changes of any other kind as long as they do not rise to the systematic, significant status that would constitute a new major version.

Users in possession of an XML file should be able to determine the version from that file, so the policy stipulates that wherever possible, the version number should be recorded internally within the file to which it applies. In TEI documents, we record the version number using the @n attribute within the <edition> element in the <editionStmnt> section of the header. We also recommend documenting the revision history of the file using <change> elements within the <revisionDesc> section of the header; the version number should be attached to each change newly introduced within a particular version using the @n attribute on the <change> entity. In this way, we can both identify particular states of files and construct a human-readable history of how the file developed. Where version numbers and change histories cannot practically be included in the file itself, we will store them in a supplementary text file to be archived and distributed with the data files.

61

Conclusion

The practices considered and developed by the PPEA offer a starting point for versioning digital projects, laying out standards for what needs to be versioned and how version numbers can make the status of files and their histories more intelligible. Other projects, with different needs, materials, data formats, and philosophies may need to develop different strategies in order to make their material comprehensible and usable. Development of standard practices would benefit the field of digital editing as a whole. And standards and mechanisms for versioning will have to continue to evolve alongside ongoing developments in the field of digital editing. The versioning protocol developed for the PPEA is based on a document-driven paradigm of the digital edition — that is, on a model in which key informational components of the edition are contained in individual XML files, to which version numbers can be attached. But this model is a notably simple one, even in the context of the TEI. XML documents need not be self-contained; an XML document can virtually include content drawn dynamically from other sources, a design pattern that Alan Liu terms “data pours” and finds characteristic of the modern web [Liu 2004, 59–63]. Each source file could be versioned individually, but the compound XML that is processed to render the edition may never before have existed as a coherent whole; such hybrid documents threaten a nearly endless proliferation of versions, not to mention challenging technical measures to expose the version number of each element. And more complicated paradigms may become increasingly common in sophisticated digital editions of the future. For instance, in editions developed according to the principles of “stand-off” markup, there may be a “source” document containing a core stream of textual data, designed to work in tandem with various kinds of markup stored outside that document [TEI Consortium 2019, §16.9].^[46] Some editions may even avoid storing data in documents that map to a traditional file system, opting instead for databases or other complex storage structures.^[47] Versioning practices will require ongoing consideration to keep pace with the shifting field. These are conversations the digital editing community needs to begin to have.

62

Until even an initial community consensus emerges, individual projects will have to develop their own approaches to versioning their data — approaches that will shift in tandem with their needs, infrastructures, material, and scale. Accordingly, instead of a set of rules, I conclude with three principles that can help to guide discussions about versions of texts:

63

1. Digital editions must version their underlying data and communicate those versions to users, independent of how that data is displayed. This is not the same as tracking file history in version control; nor is it the same as the bibliographical analysis scholars of future generations may want to perform. It is a declarative act in which digital editors make assertions about the state of their work. Where editorial projects offer reading interfaces or APIs, they should strongly consider versioning their software environments, due to the complicated technological interactions required to display a text. However, versioning the data itself is of paramount importance. Wherever possible, editions should provide direct access to their versioned data (for example, in the form of TEI-encoded XML files) so that users can examine the data directly, apart from its interface.
2. Versioning is social. As debates in the software community have suggested, versioning is not an abstract concept, but is inherently tied to *use*. Developing versioning principles will require editorial projects to have a use-model of their resources, one that takes into account what kinds of changes are intellectually and practically significant. This means, for instance, deciding what types of object are fundamental to the resource and at what level they should be versioned. (A single epigraph, or a corpus? Chapter or

novel? Poem or volume? The entire archive offered by a large project? Given both creators and users, how should we understand the resource as transforming?)

3. Digital editions must explicitly scope their revisions, delivering version numbers that communicate with users (based on their needs) the scale and significance of the change. It should be possible to understand through version numbers not only what version of a resource is most recent, but how “compatible” they are, how likely it is that the differences have a significant impact on their intellectual coherence or their probable uses. Both in individual projects and the field of digital editing as a whole, we should develop explicit guidelines that make these versions meaningful.

Kirschenbaum has claimed that despite the significant technical challenges of digital preservation, its greatest challenges are “ultimately — and profoundly — social” [Kirschenbaum 2008, 21]. The same, I would argue, is true of the issues surrounding the evolution and internal histories of digital editions. The field must begin to develop standards and practices for managing resource histories — standards and practices that ideally should not be limited to any one file format or encoding scheme, but can help organize data of many forms, for many purposes, now and in the future. And other practices will need to develop around those standards: for instance, support for versioning in emerging APIs for digital scholarly text. Version numbers, I argue, can help meet these needs, and it is time for digital editors to begin discussing and using them.

64

Acknowledgements

My thanks to Timothy Stinson for his feedback on this article, and to Matthew Kirschenbaum for his comments on an earlier version of this work. I am also grateful to my many generous interlocutors at the BH and DH conference where I first presented these ideas, and to anonymous reviewers at DHQ. This research was made possible by the support of a CLIR Postdoctoral Fellowship in Data Curation for Medieval Studies at the North Carolina State University Libraries. Another version of this article will be published as “Versioning and Digital Editions”, in *Book History and Digital Humanities*, Wacha, H. and Vareschi, M. (eds), Center for the History of Print and Digital Culture and University of Wisconsin Press, Madison, forthcoming.

65

Appendix 1

To get an idea of existing versioning practices in existing digital editions, I examined versioning and revision-tracking practices in the thirty “interesting editions/projects” singled out for recommendation in Patrick Sahle’s online *Catalogue of Digital Editions* [Sahle 2019]. These resources, comprising projects with start dates ranging from 1995 to 2018 and covering multiple fields, languages, and kinds of material, offer a convenient sampling of available digital editorial work. I examined each resource and attempted to determine whether it provided version numbers and whether it kept granular revision histories. To attempt to learn a project’s practices, I examined its opening page, pages describing the project and its technical and editorial policies, credits pages, citation instructions, and a small sampling of pages displaying texts belonging to the edition. Where a project provided direct access to its underlying data files (typically in the form of TEI-encoded XML), I also examined a few of these files to see if version numbers or revision histories were represented in the data files. Because my examination of the data files of any individual project was limited and manual, it is possible that a project includes version information or revision history in a file I did not examine, or discusses these matters in a portion of the site I did not access; however, my examination suggests broadly whether such information is accessible to site users. My findings are summarized in the table below, followed by brief discussion.

66

Project Name ^[48]	Project-wide version number	Version numbers for individual documents	Change logs in data files ^[49]	Other detailed change logs	Provides date of last update ^[50]
Jane Austen's Fiction Manuscripts Digital Edition					
Bayeux-Tapestry Digital Edition ^[51]					
Samuel Beckett - Digital Manuscript Project ^[52]			X		
Burckhardt Source					
Lord Byron and his Times		X			
The Canterbury Tales Project: The Miller's Tale on CD-ROM					
The Canterbury Tales Project: The Nun's Priest's Tale on CD-ROM					
Dante Alighieri: Commedia - A Digital Edition ^[53]					
Alfred Escher - Briefedition			X		
Faustedition / Johann Wolfgang Goethe: Faust. Historisch-kritische Edition			X		
In Transition: Selected Poems by the Baroness Elsa von Freytag-Loringhoven					
The Diary of William Godwin			X		
The Thomas Gray (1716-1771) Interactive Online Commentary				^[54]	X
The Charles Harpur Critical Archive					
Wolfgang Koeppen: Jugend - Textgenetische Edition					
Hugo von Montfort - Das poetische Werk					
The Newton Project			X		
The Proceedings of the Old Bailey, London 1674 to 1834	X			X	
Petrus Plaoul - Editio Critica Commentarii in libris Sententiarum ^[55]		X			
The Complete Writings and Pictures of Dante Gabriel Rossetti - A Hypermedia Archive		X			
Arthur Schnitzler - Digitale historisch-kritische Edition (Werke 1905 bis 1931)	X			X	
Codex Sinaiticus		X	X		
Bichitra: Online Tagore Variorum					X
Digital Thoreau					
Vincent van Gogh - The Letters					X ^[56]
Van Nu en Straks. De Brieven			X		
Lope de Vega - La Dama Boba - EDICIÓN CRÍTICA Y ARCHIVO DIGITAL					
The Digital Vercelli Book	^[57]			X ^[58]	
Carl-Maria-von-Weber-Gesamtausgabe (WeGA) [Digitale Präsentation]	X		X		
The Walt Whitman Archive			X	X	
Total Number	3	4	9	4	3

Table 1.

Of the projects represented, 18 (60%) acknowledge in some form that their resources may change over time. The most common form of acknowledging changes, practiced by thirteen projects (43%), is maintain a change log, which typically records a description of changes, the date on which they were made, and who made them. Four projects provide a list of changes as part of the website (in the case of the *Vercelli*, I suspect generated from the underlying data files, which are not accessible). These vary in detail; *Schnitzler* focuses on the addition of new content and features; *Old Bailey* discusses corrections but often summarizes changes made to many records simultaneously; the *Whitman Archive* provides a detailed description of revisions in an external blog. Ten projects, all encoded in TEI XML or a derivative format, store revision lists in each data file

using the `<revisionDesc>` element; the *Whitman Archive* is noteworthy in providing both internal and external change logs.

Supplying version numbers is a rarer approach. Seven projects (23%) employ version numbers in some capacity. This total is split almost evenly between projects that assign a single version number to a given state of the project (3; 10% of the total) and those that version separate texts or data files independently (4; 13% of the total). Surprisingly, only three projects combine version numbers with a detailed listing of changes; in all cases one version number applies to the resources as a whole, though one of the projects records changes to individual files while the other two announce sitewide changes (including new features) in tandem with new versions.

68

The three projects using sitewide version numbers all use conventional two-segment version numbers.^[59] *Schnitzler* is the only project to explicitly explain the meaning of its version numbers, the two segments of which correspond to major and minor releases: major releases are marked by the release of significant new functionality or materials, as anticipated by the phases mapped out by the Release Plan, while minor versions correspond to minor updates [Informationen zum Beta-release 2.0 2019].^[60] Generally, these project-wide version numbers assign clear version numbers that clearly communicate something about the scope of their changes.

69

By contrast, the projects that grant version numbers to individual documents or data files use a much wider array of formats. Of the four projects that version individual resources, two conceive of versioning by analogy to print. The *Rossetti Archive*, on the pages for individual works within the archive, refers not to versions of its materials but to editions. At the bottom of the page for each item in the archive, the site gives an “Electronic Archive Edition” number as an integer; an item might be listed as edition 1 or 2.^[61] Inspecting the XML files reveals that the book analogy is suggested in part by the TEI; the edition number is given in the document header using the `<edition>` element.^[62] (*Lord Byron* similarly attaches an integer edition number to the `<edition>` element, though it uses the `@n` attribute, where *Rossetti* makes the edition number the element’s content; *Lord Byron* does not display this number in the reading interface.) These numbers provide a mechanism for recording changes, but the number’s low resolution combined with the ambiguity of the `<edition>` element and the lack of a stated versioning policy means that it is difficult to be certain the edition number will be updated with any change.

70

Petrus Plaoul, as available in the Scholastic Commentaries and Texts Archive, also refers to a state of one of its digital texts as an edition, but the identifiers it assigns suggests a more robust way of thinking about textual state. The identifier for an “edition” might take the form “2011.10-dev-master”, accompanied by the date “October 04, 2011”; these appear at the head of every text on the site (for example, Plaoul, 2011). These identifiers offer a form of Calendar Versioning, labeling a state of the text according to when it last changed, combined with what appears to be technical control information. The dot separation between the year and month visually evokes standard formats for version numbers. But of the projects profiled, only *Codex Sinaiticus* explicitly refers to states of its material as versions. The website detaches this information from the presentation of the text; I found the version number listed only on the XML Download page, where it is also accompanied by a revision date [XML Download n.d.]. The version number — 1.04 at the time of writing — is stored in the downloadable XML file, attached to the `@n` attribute of the `<edition>` element and also labeled “Version 1.04” in the content of that element, where it is accompanied by a date of last update (March 25, 2014). The `<revisionDesc>` element enumerates changes made to the XML file and parenthetically links each to the version number of the file in which the change was made. Of the projects examined, *Codex Sinaiticus* alone offers detailed version numbers capable of registering the scope of revisions, and it is also alone in explicitly articulating the link between labeled version and the revision history.

71

The field of digital editing as a whole shows an understanding of the importance of acknowledging resource change; a majority of the editorial projects surveyed take some steps to show how the current state of the resources differs from earlier forms in which the same resource was available. Based on the prevalence of various approaches to change, it appears that at least a partial consensus has formed within the field about using change logs to describe to human readers the changes that have occurred. By comparison, labeling specific states of the resource through the use of version numbers or other identifiers is much less common, and even among projects that do explicitly version resources, practices are wildly inconsistent. Should whole websites be versioned, or individual texts? What constitutes a version? What form should version numbers take? This article has argued that version numbers are necessary for data management and interoperability among digital editions. The significant disparities in existing practices highlight the need for a field-wide conversation to develop practices around versioning practices.

72

Notes

[1] Tanselle (2001) objects to the idea that electronic texts are in any way more fluid than printed texts. Modifying electronic files, Tanselle says, alters outputs no more completely or undetectably than does the resetting of type. But Tanselle undervalues the ways in which digital textuality (especially online) collapses the space between creation and publication: a revised form is not broadcast into copies already printed, but changes made to a file on a webserver will immediately appear to anyone visiting the website, even if they have previously visited it, unless they have taken pains to archive a copy, and any savvy web user understands that an online resource may not be the same as last time they visited. The print world may be following the electronic: in today’s publishing environment, books are “born digital”, designed on computer screens and printed with laser printers or with plates that are designed to be disposable and can be regularly recycled and recreated. I learned about current publishing practices from a talk by Matthew Kirschenbaum [Kirschenbaum 2017], which has considerably influenced my thinking. Today’s printed books thus share in digital instability, and their bibliographers and archivists will need to be concerned with many of the issues of digital revision and versioning that I discuss in this article.

[2] As is frequently noted — see for example Schreibman (2013), ¶41 and Sahle (2016), 29-30.

[3] These might be changes to underlying technologies, such as updating a piece of software on a webserver or upgrading to a version of a web framework designed for newer browsers, but they might also be changes in support of the long-term interoperability and accessibility of underlying data, such as migrating data to a new encoding standard after an old one has become obsolete.

[4] This definition is broadly similar to that offered in Sahle (2016). *Textual objects* is an intentionally expansive term, most obviously encompassing literary and historical works

and documents, but potentially describing any materials that could be encoded and edited. I make no distinction between “edition” and “archive” (see Price (2009)), and also refer to the organizations and publishing outlets that create and provide access to edition materials as “projects” and to their outputs as “texts” or, more generally, “resources”. The versioning problems with which I am concerned affect editions of all size, from ad-hoc encodings of individual documents to large digital archives encompassing many edited texts. While I am primarily thinking of richly encoded editions such as those based on Text Encoding Initiative standards, the issues and solutions I present would apply equally to plain text files.

[5] In addition to peer review processes offered by publishers, professional organizations have created mechanisms for peer reviewing digital editions. The Modern Language Association’s Committee on Scholarly Editions (MLA CSE) seal, awarded to Approved Editions, is available to print and digital editions alike. Member organizations of the Advanced Research Consortium — the Medieval Electronic Scholarly Alliance (MESA), 18thConnect, Nineteenth-Century Scholarship Online (NINES), ModNets, and Studies in Radicalism Online (SIRO) — also facilitate peer review processes for digital resources including digital editions. According to the ARC, when a node approves a resource, the node’s director issues a letter “geared toward tenure and promotion committees” that “highlights equivalencies to print publications” [Scholarly Peer Review n.d.].

[6] Of the thirty projects surveyed in the Appendix, only *The Proceedings of the Old Bailey* in my assessment prioritizes access to textual data over the reading of text. Turksa et al. emphasizes the degree to which editors, not to mention funders and potential publics, respond to the presentation of digital editions. The massive body of writing emphasizing the flexible displays and interfaces of digital editions implicitly understands them as resources that users will interact with through reading interfaces [Turksa et al. 2016, ¶2–5]; see for example Tanselle (1995), 591-2 and Shillingsburg (1996), 163-6. Shillingsburg suggests that digital editions are not well suited for novice or pleasure readers [Shillingsburg 1996, 165], a proposition echoed by Gabler (2010): “we read texts in their native print medium, that is, in books; but we study texts and works in editions — in editions that live in the digital medium.” However, note Krista Stinne Greve Rasmussen’s assertion that the role of reader is the foundation for more involved forms of textual study and knowledge creation [Rasmussen 2016, 128].

[7] That is, while commentators have praised the ways in which digital editions expose the editorial process and invite readers to interrogate the editors’ methods (see for example Smith (2004), 317-318; Gabler (2010), 48), digital editions still typically produce texts (even if those texts are multiple or provisional), and readers may well want to reference those texts as texts, using them as a basis for literary analysis, rather than engaging with them as arguments about text. A study of users of the Font Gaia digital library (which includes digitized content, digital exhibits, and digital editions) found that the most common use for the library was to consult documents online, though users were primarily “scanning” and reading selectively rather than reading in full — occupying, the study’s author notes, Rasmussen’s “user” role [Leblanc 2018, 295, 297, 303–304].

[8] See Kalvesmaki (2014) for a discussion the Canonical Text Services (CTS) standard for digital cross-references. The in-progress Distributed Text Services specification builds on the work of CTS to develop systems for computers to query and retrieve data from digital editions [Distributed Text Services 2019].

[9] In the Appendix I present the results of an examination of thirty digital editions, finding that only 23% give their material version numbers, and only 13% version individual data files representing distinct texts. Major works devoted to scholarly editing in the digital age also omit discussion of versioning materials post-publication. Shillingsburg (1996), 169 and Kline and Perdue (2008), 288 both comment approvingly on the ability of editors to make corrections to published electronic editions, and both devote attention to managing data during the preparation of an edition, but neither offers concrete suggestions for managing changes to materials after publication. The essays in Burnard et al. (2006), offer a good deal of practical insight into data issues of digital editions, and two address head-on the challenges posed by the mutability of digital editions [Berrie et al. 2006] [Deegan 2006], but none of the contributions suggests clear practices for publicly versioning materials. The MLA CSE’s Guidelines for Editors of Scholarly Editions ask editors to consider the importance of “permanence or fixity” as well as the benefits of “openness and fluidity” (§1.2.3), and ask those charged with vetting editions in all media to determine whether a correction file will be available (§2, questions 22.3-4), and in the case of digital editions whether edition materials have been deposited in a long-term repository (§2, question 28.4); however, the guidelines offer no standards for how digital editions should make users aware of changes or ensure long-term reference [MLA Committee 2011]. Pierazzo (2015), 186-187 directly addresses the problem of versioning, perhaps heralding a needed increase of attention toward the issue. Pierazzo’s suggestion is to embed a revision control system within a digital edition; I will discuss the limitations of that approach below.

[10] For one statement on the impossibility of a definitive edition, see Tanselle (1992), 74. On an edition as presenting a theory about the work being edited, see Cerquiglini (1999), 22.

[11] The most recent edition of the APA style guide eliminates the recommendation to provide access dates [Publication Manual 2010, §6.31ff]. By contrast, the 2003 edition of *The Chicago Manual of Style* initiated the still-standing Chicago style recommendation against access dates. It also warned against including revision dates, though that stance has since weakened [Chicago Manual 2003, §17.12].

[12] The blog also provides descriptions of additions and modifications to the *Archive* website apart from updates to the XML data, though the blog description notes that minor changes in appearance and events such as server outages are not recorded.

[13] These systems are also known as version control systems; I use the term revision control systems to emphasize the fact that these systems record project data and changes to it, but do not identify versions of the data unless those versions are explicitly labeled within the RCS. The terms are essentially interchangeable in their typical use; Git, for example, identifies itself as a “version control system” on its homepage but as a “revision control system” in its manual [Git n.d.] [Git User’s Manual n.d.].

[14] RCSs can still be useful for presenting and exploring project history even if not embedded within the edition. For one endorsement, see Escobar Varela (2016) ¶¶34-35. Release tagging, which Escobar Varela highlights, can be used in tandem with version numbering to label a particular state of the repository as representing a specific version — but this works effectively only where the contents of the repository are versioned together as a unit.

[15] It would be possible to automatically embed information regarding RCS revisions into data files so they preserved the information even if removed from the RCS, as one project profiled in the Appendix did; see note 56. However, such measures are workarounds that highlight the extent to which RCS revision numbers differ from version identifiers created specifically for a resource.

[16] I collaborated with Daniel Paul O’Donnell to use Zenodo to publish and version the source code for the online republication of his digital edition of *Cædmon’s Hymn* [O’Donnell 2018].

[17] On the prominence of documentary editing in the digital sphere, see Pierazzo (2014).

[18] I cite only a few productive samples from these wide-ranging debates. For a concise and helpful, though dated, overview, see Greetham (1992), 335-346.

[19] This is not, of course, to suggest that the edition exists apart from the history of the work edited. A new edition becomes the latest entry in the textual history of the work it edits — it might even be said to constitute a version of that text — and a future study of the reception or evolution of the work might include the edition as one of its objects of study. But for the purposes of publication and data management, we need to think of the edition as its own unit and manage its evolution.

[20] In general, the individual numbers making up each segment are independent integers, so that 3.11.15 is a valid version number with major version 3, minor version 11, revision or patch 15.

[21] The CalVer proposal was released in 2016, but as the authors note, the practices they describe predate the document. Rather than trying to impose a standard format, the CalVer convention seeks to provide a common vocabulary and expose influential practices.

[22] Ironically, at the time of writing, the Semantic Versioning specification suffers from its own versioning problems; the version cited differs in two minor details from the version available at <https://semver.org/> (archived at <http://web.archive.org/web/20190803230500/https://semver.org/>), though both are labeled as version 2.0.0.

[23] In light of this article's argument, it is worth pointing out that Ashkenas posted his manifesto to GitHub's Gist service, which versions files using Git. The document has been revised several times since its creation, and the hash in the URL allows me to link to a particular state of the document, but does not provide a way to signal how the state I cite (the most recent at the time of writing) relates to other states.

[24] For a satiric presentation of "sentimental versioning", see Tarr (n.d.).

[25] Boot and van Zundert stress the importance of versioning the individual resources within the edition-networks they imagine, and suggest that the systems for managing data and services should even handle the versioning of platform infrastructure such as the operating systems on which technical services may depend [Boot and Zundert 2011, 148].

[26] Witt (2018) argues for making APIs the foundational avenues of data access and for constructing user interfaces as applications that consume data through APIs — if adopted at scale, an elegant approach to multiplicity and reuse.

[27] See Bradley (2012) for one critique of the marginalization of collaborators with technical expertise as "techies", which Bradley argues improperly reduces technical contributions to "support work" [Bradley 2012, 11] rather than recognizing the important intellectual contributions and innovations that all partners bring to the table. Bradley specifically notes the importance of "blending of the understanding of the materials with which one is working with an understanding of how to exploit the technology to emphasize what is important" as one important area of partnership [Bradley 2012, 14]. I hope it will be clear that the separation of versioning I propose is not meant to downplay the intellectual importance of the technological components, either from the perspective of labor or from the perspective of scholarly resources.

[28] A related problem exists in the study of videogames, where many older games are experienced using emulators and researched through ROM files extracted from original media by third parties. For a discussion of the bibliographic description of such objects, see Altice (2015), 333-341, which argues among other things that videogame scholars should cite even the emulators they use to examine such files.

[29] For example, in modern web development, the accepted best practice is for the structure of the document to be specified in HTML, while formatting is applied using CSS. For a discussion of this principle, see Berners-Lee (1998). In the context of digital editing, see Pichler and Bruvik (2014).

[30] For a critique of the separation from the standpoint of textual studies, see Gale (2010), 110-114.

[31] The TEI Guidelines credit this separation as a characteristic of the XML encoding language, which emphasizes "descriptive" rather than "procedural" markup: that is, the markup categorizes pieces of a document according to what they mean or the structural purpose they serve rather than according to how they should be formatted; the formatting of a published document should be accomplished through other mechanisms [TEI Consortium 2019, §v.1]. That is not to say that the TEI Guidelines lack any facilities for describing the appearance of texts, but the Guidelines stress that components related to visual appearance are intended to describe a source document, not its desired output appearance (§1.3.1.1.3), and note that markup describing the visual features of a source document is descriptive markup (§v.2).

[32] Régnier (2014), 76, argues that "philologists can . . . be held responsible for the functional and aesthetic quality of the digital framework to which they entrust their work" and insists that "they have to collaborate on the invention of digitized text standards" like the visual codes that coalesced as standards for print scholarship.

[33] Sperberg-McQueen declares that the selection and presentation together constitute the interface of an edition. I use the term interface differently, to refer to a mechanism through which the edition exposes its information, whether displaying it visually through a graphical user interface (GUI) or exposing it to other computer programs by means of an application programming interface (API).

[34] Other platforms might require the edition's maintainers to perform some action to update the data file's derivatives, for example running a script to generate new HTML files for web display by applying an XSLT transformation to a source XML file. Only if the edition is exceptionally tightly packaged is it likely that the software of the edition must also be regenerated, and even if it is, the resulting display software will not be materially different.

[35] For an introductory overview to issues of digital preservation, see Kilbride (2016). For a discussion focused on digital editions, see Deegan (2006). In combining text with (sometimes custom-built) software interfaces, digital editions present problems closely related to those involving other electronic literature. Liu et al. (2005) argue for the value of creating an XML-based format that can make content and portions of the experience of such works available even where the full experience cannot be recreated due to the obsolescence of software or hardware.

[36] See, for example, Turska et al. (2016), which argues that encoded data are the most important output of editing projects but suggests that editors are concerned with presentation and so lowering the barriers to publication will help them get down to the business of creating data.

[37] Gants (2010), 133-134, considering the issues involved in describing a work of interactive fiction that takes the form of a computer program, proposes a similar identity. Using Bowers's bibliographic framework, Gants compares the source code for the game to a single setting of type; compiling the game into executable code that will run on separate operating systems, he suggests, is analogous to reimposition in other formats.

[38] For an overview of the importance and challenges of software preservation and curation and a discussion of the role research libraries might play, see Chassanoff et al. (2018).

[39] On the history of the PPEA, see Knowles and Stinson (2014). On its early publication practices, see Duggan and Lyman (2005).

[40] The first volume of this series was published in 2018: Burrow and Turville-Petre (2018).

[41] Publication, for our purposes, means the official appearance of an edition on the public pages of the PPEA website. Because in digital scholarship the lines between unpublished and published materials have become increasingly blurred — many editors and projects, including the PPEA, make draft materials available — it may be appropriate to version prerelease materials as well, and similar procedures could apply. However, in drawing a distinction between unpublished and published materials, I emphasize that published materials have been officially recognized as appropriate for reference and citation, so users expect to be able to rely on it. Formal versioning practices support that implicit contract with users.

[42] O'Donnell (2008) uses the example of an earlier SEENET publication of his edition of *Cædmon's Hymn* to argue that print works can be outputs of digital editing. For the current print series, volumes are produced by transforming the XML source into LaTeX markup (which might be finessed by hand to improve page layout). The LaTeX markup is compiled into a PDF, which is used to print the physical volume. The physical book is thus the product of transformations of the XML source, just like the web display. The copyright page of the print book contains a statement of the version number from which it was printed, asserting the identity between them.

[43] Inspired in part by the careful and precise distinctions suggested by Semantic Versioning, I at first attempted to theorize “breaking changes” for the digital edition, trying to identify what kinds of change would render two states of the same file “incompatible” with each other. However, I soon realized that identifying “breaking changes” requires committing to a particular theory of the digital edited text and the primary form of interface it provides. People using the edition mainly as a documentary text will have different concerns from those most interested in the editors' arguments; those studying dialect will prioritize different features from those examining scribal decoration and again from scholars interested in markup practices; readers working directly with the XML files will have a very different experience of changes from the probable majority who are reading through the mediation of a web interface. In the context of digital editing, nearly every change is potentially a breaking change for someone (a claim Ashkenas made even about many software packages).

[44] On the intellectual and technical differences between P4 and P5, see Wittern et al. (2009), which observes that one of the changes it discusses marks “a fundamental change in the relationship between textual content and markup” [Wittern et al. 2009, 285]. Automated tools were developed to aid the transition from P4 to P5, and for simple files those tools might suffice, but the differences between P4 and P5 are sufficiently significant that at the conversion requires careful assessment and may require manual intervention. Because changes of this nature are interpretative, and distributed throughout the document, they amount to a significant overhaul.

[45] However, some types of widespread changes do not rise to the level of constituting a new major version, because they are intellectually trivial and do not involve theories of the text or its encoding. One example previously encountered by the project is changing the format in which line numbers are written without changing the numbers themselves.

[46] Desmond Schmidt argues that stand-off markup is essential to interoperability and should become the dominant approach to digital editing. In stand-off markup, a resource and its markup might reside in different files, and depending on technical and procedural approaches might have to be versioned separately. Stand-off markup also complicates the notion of versioning because stand-off markup is entangled with the text to which it is applied: though the markup may be stored externally, changes to the “source” document are likely to necessitate corresponding changes to the stand-off annotations in order to maintain their relationship. Schmidt's discussion offers one possible way forward: he describes his stand-off alternative to a conventional document-based edition as a “bundle” of materials in separate files, and notes that this collection of files might be stored in a single container files [Schmidt 2015, ¶47–48]. An editorial bundle consisting of a “source” document, stand-off files, and metadata might thus be versioned as a single unit, in the same way that multiple research data files may be combined into a data package consisting as multiple files, versioned as one unit.

[47] See for example Kuczera (2016).

[48] I give the short titles provided by Sahle in order to facilitate easy cross-referencing with his list, where fuller citations and hyperlinks are available. On August 4, 2019, I used the Internet Archive Wayback Machine (<https://web.archive.org>) to archive a copy of the landing page for each site on Sahle's list, as well as all the links from that page that the Archive was able to automatically follow. That process does not preserve the sites in full, but it does establish a partial record of how the sites presented themselves when I consulted them. The archived versions may be accessed by entering the URL for each site at the Wayback Machine and navigating to the date in the site history.

[49] I recorded a project as keeping change logs in data files if I actually found such logs (for instance, in a <revisionDesc> element) or if the project's technical documentation discusses creating them.

[50] Resources offering version numbers or change logs often record the dates of the changes; this column notes only those instances where a site records the date of last update in the absence of other change information. With the exception of *Van Gogh*, the resources listed in this column provide a single date of last update for the site as a whole.

[51] I accessed the online sample of this resource. The URL button at the top of the screen, which provides a recommended citation, describes this as the “revised edition”, and gives the publication date as 2011. The Credits page provides only the original date of 2003. I have not considered that statement of a revised edition, which does not seem to be repeated elsewhere, to constitute a version identifier.

[52] Most of the materials published by this project are available only to subscribers; in addition to the project's documentation, I consulted the freely available demo versions of a few texts made available on the site.

[53] I accessed the online sample of this resource.

[54] A Website History page provides updates on additions to the site materials, and individual pages, such as the Finding Aid resource, contain their own revision histories. The About page claims that “versioned corrections and revisions of the pages take place continuously” [Huber 2019], but I have not been able to find version identifiers or retrieve old versions — though the same page notes that archived versions of the site are available upon request.

[55] The URL provided by Sahle directs to the Plaoul Commentary as housed within the Scholastic Commentaries and Texts Archive (<http://scta.lombardpress.org/text/questions/plaoulcommentary>). It is this form of the resource that I have examined.

[56] Each data file includes an XML comment at the beginning of the file that provides both the data and time when the project data was last modified and a "SVN Revision" number: presumably the revision number in the Subversion repository in which the project is stored. (Apache Subversion is a revision control system predating Git.) This comment is likely created using an automated software tool when changes are committed to the Subversion repository, and all files appear to be labeled with the time and revision number of the latest commit to the project repository as a whole. I do not count these SVN revision numbers as version numbers because, as I discuss in this article, versioning involves intention and judgment. (Moreover, the revision numbers are not displayed outside of the data files.) However, SVN revision numbers do resemble project-wide version numbers more than do Git commit hashes: in SVN, revision numbers take the form of integers and each is one greater than the previous. Accordingly, these revision numbers could be used both to identify a particular state of the project and to understand the sequence among versions of a file.

[57] The Project Info menu option describes the site release as "Second digital edition (beta 2)." However, the six listed revisions suggest a more complicated change history than this numbering expresses, so I do not count it as a meaningful site-wide version number. The "second edition" appears to refer to the platform rather than to the underlying data or to the site as a whole.

[58] Does not provide access to underlying TEI files. The list of changes accompanying the edition exists in a format that might have been generated using <change> elements in the <revisionDesc> section of the TEI header, though it is impossible to be sure without the underlying XML.

[59] At the time of writing, *Proceedings of the Old Bailey* is on version 8.0, *Schnitzler* is on beta version 2.0, and *WeGA* is on version 3.4.

[60] However, it is ambiguous whether all content changes register in the site's version history, which currently only notes the addition of a new resource in conjunction with the release of Beta 2.0. Following its explanation of its versioning practices, the site explains, "Kleinere, z.B. von Benutzern gemeldete Fehler (Bugfixing und inhaltliche Fehlerkorrekturen) werden laufend behoben" [Smaller, e.g. user-reported errors (bugfixes and content corrections) are continuously fixed]; it is unclear whether these ongoing corrections are recorded or versioned.

[61] I was able to locate only two texts in the Archive with a stated Archive Edition number of 2: Hunt (n.d.); Masterpieces of D.G. Rossetti (n.d.). For neither does the publicly available XML source include any metadata on revisions (the <revisionDesc> element is empty in both), so it is impossible to get a sense of what sorts of changes constitute a new edition. Possibly the publication of a new edition was considered to reset the revision state of the document, so that records of changes from the previous edition need not be preserved. It is also possible that the second edition resulted from creating the resources anew a second time. I looked for such materials by conducting a Google search of the rossettiarchive.org domain for the phrase "Electronic Archive Edition: 2" (and higher numbers).

[62] The *Rossetti Archive* is not actually encoded in TEI, which its creators found insufficient for the needs of their materials [McGann 2001, 89–90]. However, the *Archive's* encoding principles drew upon the standards of the TEI, and the <editionStmnt> element of the file header is one component derived from the TEI. Early version of the TEI Guidelines are available at <https://tei-c.org/Vault/Vault-GL.html>. For the earliest version of the <editionStmnt> recommendation readily available online, see Sperberg-McQueen and Burnard (1999), §5.2.2.

Works Cited

- Abilgaard 2015** Abilgaard, N.R. (2015) "On Versioning". 5 February. Available at: <http://blog.hypesystem.dk/on-versioning>. Archived at: <http://web.archive.org/web/20190803210015/http://blog.hypesystem.dk/on-versioning>.
- Altice 2015** Altice, N. (2015) *I am Error: The Nintendo Family Computer / Entertainment System Platform*. MIT Press, Cambridge, MA.
- Andrews and Zundert 2018** Andrews, T.L. and Zundert, J.J.V. (2018) "What Are You Trying to Say? The Interface as an Integral Element of Argument". In Bleier, R. et al. (eds), *Digital Scholarly Editions as Interfaces*, Books on Demand, Norderstedt, pp. 3-33. urn:nbn:de:hbz:38-91064. Available at: <http://kups.ub.uni-koeln.de/id/eprint/9106>.
- Ashkenas 2015** Ashkenas, J. (2015) "Why Semantic Versioning Isn't". GitHub Gist. Available at: <https://gist.github.com/jashkenas/cbd2b088e20279ae2c8e/39e281d9bc8f3090dd79347b5271aa9b30c2bb8b>. Archived at: <http://web.archive.org/web/20190803210633/https://gist.github.com/jashkenas/cbd2b088e20279ae2c8e/39e281d9bc8f3090dd79347b5271aa9b30c2bb8b>.
- Australian National Data Service n.d.** Australian National Data Service (n.d.) "Data Versioning". Australian National Data Service. Available at: <https://www.ands.org.au/working-with-data/data-management/data-versioning> (viewed 1 September 2017). Archived at: <http://web.archive.org/web/20190803211318/https://www.ands.org.au/working-with-data/data-management/data-versioning>.
- Berners-Lee 1998** Berners-Lee, T. (1998) "Web Architecture from 50,000 Feet". W3C. September. Available at: <https://www.w3.org/DesignIssues/Architecture.html>. Archived at: <http://web.archive.org/web/20190803211651/https://www.w3.org/DesignIssues/Architecture.html>.
- Berrie et al. 2006** Berrie, P. et al. (2006) "Authenticating Electronic Editions". In Burnard et al. 2006, pp. 269-276.
- Boot and Zundert 2011** Boot, P. and Zundert, J. (2011) "The Digital Edition 2.0 and The Digital Library: Services, not Resources". In Fritze, C. et al (eds), *Digitale Edition und Forschungsbibliothek: Beirträge der Fachtagung im Philosophicum der Universität Mainz am 13. und 14 Januar 2011*, Harrassowitz, Wiesbaden, pp. 141-152.
- Bowers 2005** Bowers, F. (2005) *Principles of Bibliographical Description*. Oak Knoll Press, New Castle, DE. First published 1949.
- Bradley 2012** Bradley, J. (2012) "No Job for Techies: Technical Contributions to Research in the Digital Humanities". In Deegan, M. and McCarty, W. (eds), *Collaborative Research in the Digital Humanities*, Ashgate, Farnham, Surrey, pp. 11-26.
- Bryant 2002** Bryant, J. (2002) *The Fluid Text: A Theory of Revision and Editing for Book and Screen*. University of Michigan Press, Ann Arbor.
- Burnard et al. 2006** Burnard, L., O'Brien O'Keefe, K., and Unsworth, J, eds. (2006) *Electronic Textual Editing*. Modern Language Association, New York.

- Burrow and Turville-Peter 2018** Burrow, J.A. and Turville-Peter, T., eds. (2018) *Piers Plowman: The B-Version Archetype (Bx)*. Society for Early English and Norse Electronic Texts, Raleigh, NC.
- Cerquiglini 1999** Cerquiglini, B. (1999) *In Praise of the Variant: A Critical History of Philology*. Johns Hopkins University Press, Baltimore.
- Chassanoff et al. 2018** Chassanoff, A. et al. (2018) "Software Curation in Research Libraries: Practice and Promise". *Journal of Librarianship and Scholarly Communication*, 6 (1). <http://dx.doi.org/10.7710/2162-3309.2239>.
- Chicago Manual 2003** *The Chicago Manual of Style* (2003) 15th ed. University Of Chicago Press, Chicago.
- Chicago Manual 2017** *The Chicago Manual of Style* (2017) 17th ed. University of Chicago Press, Chicago.
- Deegan 2006** Deegan, M. (2006) "Collection and Preservation of an Electronic Edition". In Burnard et al. 2006, pp. 358-370.
- Distributed Text Services 2019** "Distributed Text Services (DTS)" (2019). Distributed Text Services. Available at: <https://distributed-text-services.github.io/specifications/> (viewed 3 August 2019).
- Duggan and Lyman 2005** Duggan, H.N. and Lyman, E.W. (2005) "A Progress Report on *The Piers Plowman Electronic Archive*". *Digital Medievalist*, 1. <http://dx.doi.org/10.16995/dm.5>.
- Duggan et al. 2019** Duggan, H.N., Stinson, T.L., and Turville-Petre, T., eds. *Piers Plowman Electronic Archive*. Society for Early English and Norse Electronic Texts. Available at: <http://piers.chass.ncsu.edu>.
- Eaves et al. 2017** Eaves, M., Essick, R.N., and Viscomi, J., eds. (2017). *The William Blake Archive*. Chapel Hill, NC. Available at: <http://www.blakearchive.org/>.
- Eggert 2013** Eggert, P. (2013) "Apparatus, Text, Interface: How to Read a Printed Critical Edition". In Fraistat, N. and Flanders, J. (eds), *The Cambridge Companion to Textual Scholarship*, Cambridge University Press, Cambridge, pp. 97-118.
- Escobar Varela 2016** Escobar Varela, M. (2016) "The Archive as Repertoire: Transience and Sustainability in Digital Archives". *Digital Humanities Quarterly*, 10(4). Available at: <http://digitalhumanities.org/dhq/vol/10/4/000269/000269.html>. Archived at: <http://web.archive.org/web/20190804175354/http://digitalhumanities.org/dhq/vol/10/4/000269/000269.html>.
- Fitzpatrick 2011** Fitzpatrick, K. (2011) *Planned Obsolescence: Publishing, Technology, and the Future of the Academy*. New York University Press, New York.
- Folsom and Price n.d.** *The Walt Whitman Archive*. Center for Digital Research in the Humanities, University of Nebraska-Lincoln, Lincoln, NE. Available at: <https://whitmanarchive.org>.
- Franzini et al. 2019** Franzini, G., Terras, M. and Mahony, S. (2019) "Digital Editions of Text: Surveying User Requirements in the Digital Humanities". *Journal on Computing and Cultural Heritage*, 12(1). <http://dx.doi.org/10.1145/3230671>.
- Fyfe 2012** Fyfe, P. (2012) "Electronic Errata: Digital Publishing, Open Review, and the Futures of Correction". In Gold, M.K. (ed), *Debates in the Digital Humanities*, University of Minnesota Press, Minneapolis, pp. 259-280.
- Gabler 2010** Gabler, H.W. (2010) "Theorizing the Digital Scholarly Edition". *Literature Compass*, 7, 43-56. <http://dx.doi.org/10.1111/j.1741-4113.2009.00675.x>.
- Galey 2010** Galey, A. (2010) "The Human Presence in Digital Artifacts". In McCarthy, W. (ed), *Text and Genre in Reconstruction: Effects of Digitization on Ideas, Behaviours, Products and Institutions*, Open Book Publishers, Cambridge, 93-117. Available at: <https://www.openbookpublishers.com/reader/64/#page/104/mode/2up>.
- Gants 2010** Gants, D.L. (2010) "Descriptive Bibliography and Electronic Publication". *Essays and Studies*, 2010, 121-141.
- Gibaldi 1995** Gibaldi, J. (1995) *MLA Handbook for Writers of Research Papers*. Modern Language Association of America, New York.
- Gibaldi 1998** Gibaldi, J. (1998) *MLA Style Manual and Guide to Scholarly Publishing*. Modern Language Association of America, New York.
- Git User's Manual n.d.** *The Git User's Manual*. (n.d.) Version 2.22.0. Git. Available at: <https://git-scm.com/docs/user-manual.html>. Archived at: <http://web.archive.org/web/20190803235925/https://git-scm.com/docs/user-manual.html>.
- Git n.d.** "Git". (n.d.) Git. Available at: <https://git-scm.com> (viewed 22 July 2019). Archived at: <http://web.archive.org/web/20190803215454/https://git-scm.com/>.
- Greetham 1992** Greetham, D.C. (1992) *Textual Scholarship: An Introduction*. Garland, New York.
- Hashemi 2019** Hashemi, M. (2019) "Calendar Versioning". CalVer: Timely Software Versioning. 1 July. Available at: <http://calver.org>. Archived at: <http://web.archive.org/web/20190803215814/http://calver.org/>.
- Huber 2019** Huber, A. (2019) "About". *Thomas Gray Archive*. 2 July. Available at: <http://www.thomasgray.org/about/index.shtml>. Archived at: <http://web.archive.org/web/20190803220008/https://www.thomasgray.org/about/index.shtml>.
- Hunt n.d.** Hunt, W.H. (n.d.) "Pre-Raphaelitism and the Pre-Raphaelite Brotherhood". 2nd Archive Edition. In McGann, J.J. (ed), *The Complete Writings and Pictures of Dante Gabriel Rossetti: A Hypermedia Archive*, Rossetti Archive. Available at: <http://www.rossettiarchive.org/docs/nd467.h9.1914.1.rad.html>. Archived at: <http://web.archive.org/web/20190803221443/http://www.rossettiarchive.org/docs/nd467.h9.1914.1.rad.html>.
- Informationen zum Beta-release 2.0 2019** "Informationen zum Beta-release 2.0". (2019) 17 April. *Arthur Schnitzler digital*. Beta 2.0. Available at: <https://www.arthur-schnitzler.de/edition/beta>. Archived at: <http://web.archive.org/web/20190803221659/https://www.arthur-schnitzler.de/edition/beta>.
- Jones et al. 2016** Jones, S.M. et al. (2016) "Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content". *PLoS One*,

- Kalvesmaki 2014** Kalvesmaki, J. (2014) "Canonical References in Electronic Texts: Rationale and Best Practices". *Digital Humanities Quarterly*, 8(2). Available at: <http://www.digitalhumanities.org/dhq/vol/8/2/000181/000181.html>. Archived at: <https://web.archive.org/web/20190803222140/http://www.digitalhumanities.org/dhq/vol/8/2/000181/000181.html>.
- Kilbride 2016** Kilbride, W. (2016) "Saving the Bits: Digital Humanities Forever?" In Schreibman, S., Siemens, R. and Unsworth, J. (eds), *A New Companion to Digital Humanities*, Wiley Blackwell, Malden, MA, pp. 408-419.
- Kirschenbaum 2002** Kirschenbaum, M.G. (2002) "Editing the Interface: Textual Studies and First Generation Electronic Objects". *TEXT*, 14, 15-51.
- Kirschenbaum 2008** Kirschenbaum, M.G. (2008) *Mechanisms: New Media and the Forensic Imagination*. MIT Press, Cambridge, MA.
- Kirschenbaum 2017** Kirschenbaum, M.G. (2017) "Post Scripts: Graphologies of Bookmaking after Adobe". Paper presented to BH & DH: Book History and Digital Humanities, Madison, WI, 22 September.
- Klein et al. 2014** Klein, M. et al. (2014) "Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot". *PLoS One*, 9, e115253. <https://doi.org/10.1371/journal.pone.0115253>.
- Kline and Perdue 2008** Kline, M.J. and Perdue, S.H. (2008) *A Guide to Documentary Editing*. University of Virginia Press, Charlottesville.
- Knowles and Stinson 2014** Knowles, J. and Stinson, T. (2014) "The Piers Plowman Electronic Archive on the Web: An Introduction". *The Yearbook of Langland Studies*, 28, 225-238.
- Kuczera 2016** Kuczera, A. (2016) "Digital Editions beyond XML – Graph-based Digital Editions". *Proceedings of the 3rd Histo-Informatics Workshop, Krakow, Poland, 11 July 2016*, 37-46. Available at: http://ceur-ws.org/Vol-1632/paper_5.pdf.
- Leblanc 2018** Leblanc, E. (2018) *Design of a Digital Library Interface from User Perspective, and its Consequences for the Design of Digital Scholarly Editions: Findings of the Fonte Gaia Questionnaire*. In Bleier et al. (eds), *Digital Scholarly Editions as Interfaces*, Books on Demand, Norderstedt, pp. 287-315. urn:nbn:de:hbz:38-91215 Available at: <https://kups.ub.uni-koeln.de/9121/>.
- Liu 2004** Liu, A. (2004) "Transcendental Data: Toward a Cultural History and Aesthetics of the New Encoded Discourse". *Critical Inquiry*, 31, 49-84.
- Liu et al. 2005** Liu, A. et al. (2005) "Born-Again Bits: A Framework for Migrating Electronic Literature". Version 1.1. 5 August. Electronic Literature Organization, Vancouver. Available at: <https://eliterature.org/pad/bab.html>. Archived at: <http://web.archive.org/web/20190803223824/https://eliterature.org/pad/bab.html>.
- Lóscio et al. 2017** Lóscio, B.F., Burle, C., and Carlegari, N., eds. (2017) "Data on the Web Best Practices: W3C Recommendation". 31 January. W3C. Available at: <https://www.w3.org/TR/2017/REC-dwbp-20170131/>. Archived at: <http://web.archive.org/web/20190803224048/https://www.w3.org/TR/2017/REC-dwbp-20170131/>.
- MLA Committee 2011** MLA Committee on Scholarly Editions (2011) "Guidelines for Editors of Scholarly Editions". 29 June. MLA, New York. Available at: <https://www.mla.org/Resources/Research/Surveys-Reports-and-Other-Documents/Publishing-and-Scholarship/Reports-from-the-MLA-Committee-on-Scholarly-Editions/Guidelines-for-Editors-of-Scholarly-Editions>. Archived at: <http://web.archive.org/web/20190803225011/https://www.mla.org/Resources/Research/Surveys-Reports-and-Other-Documents/Publishing-and-Scholarship/Reports-from-the-MLA-Committee-on-Scholarly-Editions/Guidelines-for-Editors-of-Scholarly-Editions>
- MLA Committee 2016** MLA Committee on Scholarly Editions (2016) "MLA Statement on the Scholarly Edition in the Digital Age". May. Modern Language Association of America, New York. Available at: <https://www.mla.org/content/download/52050/1810116/rptCSE16.pdf>. Archived at: <http://web.archive.org/web/20190803225220/https://www.mla.org/content/download/52050/1810116/rptCSE16.pdf>.
- Masterpieces of D.G. Rossetti n.d.** *Masterpieces of D. G. Rossetti (1828-1882): Sixty Reproductions of Photographs from the Original Oil-paintings*. (n.d.) 2nd Archive Edition. In McGann, J.J. (ed), *The Complete Writings and Pictures of Dante Gabriel Rossetti: A Hypermedia Archive*, Rossetti Archive. Available at: <http://www.rossettiarchive.org/docs/ac-gowans.759.2r735m393.rad.html>.
- McDonough et al. 2010** McDonough, J. et al. (2010) "Twisty Little Passages Almost All Alike: Applying the FRBR Model to a Classic Computer Game". *Digital Humanities Quarterly*, 4(2). Available at: <http://www.digitalhumanities.org/dhq/vol/4/2/000089/000089.html>. Archived at: <https://web.archive.org/web/20190803224557/http://www.digitalhumanities.org/dhq/vol/4/2/000089/000089.html>.
- McGann 1995** McGann, J. (1995) "The Rationale of Hypertext". 6 May. IATH WWW Server. Available at: <http://www2.iath.virginia.edu/public/jjm2f/rationale.html>. Archived at: <http://web.archive.org/web/20190803224726/http://www2.iath.virginia.edu/public/jjm2f/rationale.html>.
- McGann 1996** McGann, J. (1996) "The Rationale of HyperText". *TEXT*, 9, 11-32.
- McGann 2001** McGann, J. (2001) *Radiant Textuality: Literature after the World Wide Web*. Palgrave, New York.
- Nielsen 2017** Nielsen, L.H. (2017) "Zenodo now supports DOI versioning!" 30 May. Zenodo Blog. Available at: <http://blog.zenodo.org/2017/05/30/doi-versioning-launched/>. Archived at: <http://web.archive.org/web/20190803225348/http://blog.zenodo.org/2017/05/30/doi-versioning-launched/>.
- O'Donnell 2008** O'Donnell, D.P. (2008) "Resisting The Tyranny of the Screen, or, Must a Digital Edition be Electronic?" *The Heroic Age*, 11. Available at: <https://www.heroicage.org/issues/11/em.php>. Archived at: <http://web.archive.org/web/20190803225437/https://www.heroicage.org/issues/11/em.php>.
- O'Donnell 2018** O'Donnell, D.P., ed. (2018) *Cædmon's Hymn: A Multimedia Study, Edition, and Archive*. Internet Edition [source code]. Version 1.1. 21 April. Zenodo. <https://doi.org/10.5281/zenodo.1226549>.
- Oxford Text Archive n.d.** The Oxford Text Archive (n.d.) University of Oxford, Oxford. Available at: <https://ota.ox.ac.uk>.
- Pichler and Bruvik 2014** Pichler, A. and Bruvik, T.M. (2014) "Digital Critical Editing: Separating Encoding from Presentation". In Apollon, D., Bélisle, C. and Régnier, P. (eds), *Digital Critical Editions*, University of Illinois Press, Urbana, Chicago, and Springfield, pp. 179-199.

- Pierazzo 2014** Pierazzo, E. (2014) "Digital Documentary Editions and the Others". *Scholarly Editing*, 35. Available at: <http://scholarlyediting.org/2014/essays/essay.pierazzo.html>. Archived at: <http://web.archive.org/web/20190803225830/http://scholarlyediting.org/2014/essays/essay.pierazzo.html>.
- Pierazzo 2015** Pierazzo, E. (2015) *Digital Scholarly Editing: Theories, Models and Methods*. Routledge, London. <https://doi.org/10.4324/9781315577227>.
- Plaoul 2011** Plaoul, P. (2011) "Lectio 14, De Fide". 2011.10-dev-master. 4 October. Witt, J.C. (ed), *Scholastic Commentaries and Texts Archive*, LombardPress. Available at: <http://scta.lombardpress.org/text/lectio14>. Archived at: <http://web.archive.org/web/20190803230348/http://scta.lombardpress.org/text/lectio14>.
- Preston-Werner n.d.** Preston-Werner, T. (n.d.) "Semantic Versioning". Version 2.0.0. Available at: <https://semver.org/spec/v2.0.0.html>. Archived at: <http://web.archive.org/web/20190803230451/https://semver.org/spec/v2.0.0.html>.
- Price 2009** Price, K.M. (2009) "Edition, Project, Database, Archive, Thematic Research Collection: What's in a Name?" *Digital Humanities Quarterly*, 3(3). Available at: <http://www.digitalhumanities.org/dhq/vol/3/3/000053/000053.html>. Archived at: <http://web.archive.org/web/20190803231004/http://www.digitalhumanities.org/dhq/vol/3/3/000053/000053.html>.
- Publication Manual 2001** *Publication Manual of the American Psychological Association* (2001) American Psychological Association, Washington, DC.
- Publication Manual 2010** *Publication Manual of the American Psychological Association* (2010) American Psychological Association, Washington, DC.
- Rasmussen 2016** Rasmussen, K.S.G. (2016) "Reading or Using a Digital Edition? Reader Roles in Scholarly Editions". In Driscoll, M.J. and Pierazzo, E. (eds), *Digital Scholarly Editing: Theories and Practices*, Open Book Publishers, Cambridge, pp. 119-133. <http://dx.doi.org/10.11647/OBP.0095.07>.
- Reiman 1987** Reiman, D.H. (1987) "'Versioning': The Presentation of Multiple Texts". In *Romantic Texts and Contexts*, University of Missouri Press, Columbia, pp. 167-180.
- Robinson 2013** Robinson, P. (2013) "What Digital Humanists Don't Know about Scholarly Editing; What Scholarly Editors Don't Know about the Digital World". Paper presented to Social, Digital, Scholarly Editing, University of Saskatchewan, 11-13 July. Available at: https://www.academia.edu/4124828/SDSE_2013_why_digital_humanists_should_get_out_of_textual_scholarship.
- Régnier 2014** Régnier, P. (2014) "Ongoing Challenges for Digital Critical Editions". In Apollon, D., Bélisle, C. and Régnier, P. (eds), *Digital Critical Editions*, University of Illinois Press, Urbana, pp. 58-80.
- Sahle 2016** Sahle, P. (2016) "What is a Scholarly Digital Edition?" In Driscoll, M.J. and Pierazzo, E. (eds), *Digital Scholarly Editing: Theories and Practices*, Open Book Publishers, Cambridge, pp. 19-40. <http://dx.doi.org/10.11647/obp.0095.02>.
- Sahle 2019** Sahle, P. (2019) "Some particularly interesting editions/projects". In *A Catalog of Digital Scholarly Editions*. Version 3.0, snapshot 2008ff. 19 February. Available at: http://www.digitale-edition.de/vlet_interesting.html. Archived at: http://web.archive.org/web/20190803232312/http://www.digitale-edition.de/vlet_interesting.html.
- Schmidt 2015** Schmidt, D. (2014) "Towards an Interoperable Digital Scholarly Edition". *Journal of the Text Encoding Initiative*, 7. doi: 10.4000/jtei.979.
- Scholarly Peer Review n.d.** "Scholarly Peer Review". (n.d.) ARC. Available at: <http://ar-c.org/about/peer-review/> (viewed 15 July 2019). Archived at: <http://web.archive.org/web/20190803233452/http://ar-c.org/about/peer-review/>.
- Schreibman 2013** Schreibman, S. (2013) "Digital Scholarly Editing". In Price, K.M. and Siemens, R. (eds), *Literary Studies in the Digital Age: An Evolving Anthology*, MLA Commons, New York. Available at: <https://dlsanthology.mla.hcommons.org/digital-scholarly-editing/>. Archived at: <http://web.archive.org/web/20190803233920/https://dlsanthology.mla.hcommons.org/digital-scholarly-editing/>.
- Shillingsburg 1991** Shillingsburg, P.L. (1991) "Text as Matter, Concept, and Action". *Studies in Bibliography*, 44, 31-82.
- Shillingsburg 1996** Shillingsburg, P.L. (1996) *Scholarly Editing in the Computer Age*. University of Michigan Press, Ann Arbor.
- Siemens et al. n.d.** Siemens, R. et al., eds. (n.d.) *A Social Edition of the Devonshire MS (BL Add. MS 17492)*. WikiBooks. Available at: https://en.wikibooks.org/wiki/The_Devonshire_Manuscript.
- Smith 2004** Smith, M.N. (2004) "Electronic Scholarly Editing". In Schreibman, S., Siemens, R. and Unsworth, J. (eds) *A Companion to Digital Humanities*, Blackwell, Malden, MA, pp. 306-322.
- Smith et al. 2016** Smith, A.M. et al. (2016) "Software Citation Principles". *PeerJ Computer Science*, 2:e86. doi: 10.7717/peerj-cs.86.
- Sperberg-McQueen 2009** Sperberg-McQueen, C.M. (2009) "How to Teach Your Edition How to Swim". *Literary and Linguistic Computing*, 24, 27-39. <http://dx.doi.org/10.1093/lilc/fqn034>.
- Sperberg-McQueen and Burnard 1999** Sperberg-McQueen, C.M. and Burnard, Lou, eds. (1999) *Guidelines for Electronic Text Encoding and Interchange*. Revised Reprint. May. TEI P3 Encoding Initiative, Chicago, Oxford. Available at: <https://tei-c.org/Vault/GL/P3/index.htm>.
- TEI Consortium 2019** TEI Consortium (2019) *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative. Version 3.6.0, revision daa3cc0b9. 16 July. Available at: <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html>.
- Tanselle 1975** Tanselle, G.T. (1975) "The Bibliographical Concepts of Issue and State". *Papers of the Bibliographical Society of America*, 69, 17-66.
- Tanselle 1980** Tanselle, G.T. (1980) "The Concept of Ideal Copy". *Studies in Bibliography*, 33, 18-53.
- Tanselle 1992** Tanselle, G.T. (1992) *A Rationale of Textual Criticism*. University of Pennsylvania Press, Philadelphia.
- Tanselle 1995** Tanselle, G.T. (1995) "Critical Editions, Hypertexts, and Genetic Criticism". *Romanic Review*, 86, 582-593.

- Tanselle 2001** Tanselle, G.T. (2001) "Thoughts on the Authenticity of Electronic Texts". *Studies in Bibliography*, 54, 133-136.
- Tarr n.d.** Tarr, D. (n.d.) "Sentimental Versioning, Version One dot Oh, okay then". Available at: <http://sentimentalversioning.org> (viewed 15 July 2018). Archived at: <http://web.archive.org/web/20190803234954/http://sentimentalversioning.org/>.
- TextGrid Consortium 2006-2014** TextGrid Consortium (2006-2014) *TextGrid: A Virtual Research Environment for the Humanities*. TextGrid Consortium, Göttingen. Available at: <https://textgrid.de>.
- Turabian et al. 1996** Turabian, K.L., Grossman, J.B. and Bennett, A.B. (1996) *A Manual for Writers of Term Papers, Theses, and Dissertations*. 6th ed. University of Chicago Press, Chicago.
- Turska et al. 2016** Turska, M., Cummings, J. and Rahtz, S. (2016) "Challenging the Myth of Presentation in Digital Editions". *Journal of the Text Encoding Initiative*, 9. <http://dx.doi.org/10.4000/jtei.1453>.
- Walt Whitman Archive Changelog 2019** "Walt Whitman Archive Changelog". (2019) Blogger. Available at: <http://wwa-changelog.blogspot.com> (viewed 3 August 2019).
- Witt 2018** Witt, J.C. (2018) "Digital Scholarly Editions and API Consuming Applications". In Bleier, R. et al. (eds) *Digital Scholarly Editions as Interfaces*, Books on Demand, Norderstedt, pp. 219-247. urn:nbn:de:hbz:38-91182. Available at: <http://kups.ub.uni-koeln.de/id/eprint/9118>.
- Witt n.d.** Witt, J.C., ed. (n.d.) The SCTA Reading Room. Scholastic Commentaries and Texts Archive. LombardPress. Available at: <http://scta.lombardpress.org> (viewed 3 August 2019).
- Wittern 2013** Wittern, C. (2013) "Beyond TEI: Returning the Text to the Reader". *Journal of the Text Encoding Initiative*, 4. doi: 10.4000/jtei.691.
- Wittern et al. 2009** Wittern, C., Ciula, A. and Tuohy, C. (2009) "The Making of TEI P5". *Literary and Linguistic Computing*, 24(3), 281-296. <http://dx.doi.org/10.1093/lc/fqp017>.
- XML Download n.d.** "XML Download of the Electronic Transcription of Codex Sinaiticus" (n.d.) *Codex Sinaiticus*, Codex Sinaiticus Project. Available at: http://codexsinaiticus.org/en/project/transcription_download.aspx (viewed 1 August 2019). Archived at: http://web.archive.org/web/20190801161134/http://www.codexsinaiticus.org/en/project/transcription_download.aspx.
- Zeller 1975** Zeller, H. (1975) "A New Approach to the Critical Constitution of Literary Texts". *Studies in Bibliography*, 28, 231-264.