DHQ: Digital Humanities Quarterly
2022
Volume 16 Number 2

Hidden in Plain-TeX: Investigating Minimal Computing Workflows

Nabeel Siddiqui <siddiqui_at_susqu_dot_edu`>, Susquehanna University https://orcid.org/0000-0002-6126-5833

Abstract

Drawing on software studies, data feminism, digital rhetoric studies, information science, and the history of computing, this paper foregrounds Markdown as a cultural object to analyze the social, cultural, and political pressures surrounding the digital humanities. Rather than beginning with contemporary discourse, it draws parallels between Markdown and Donald Knuth's TeX. From the 1970s to 1990s, academic researchers used TeX to construct plain-text scholarship in mathematics and the hard sciences to enhance typography. Most academics saw these concerns as holding marginal importance and quickly abandoned the approach for WYSIWYG word processors. Drawing on queer theorist Michael Warner, I argue that the community surrounding TeX responded reactionarily to these transformations by forming a counterpublic constituted through a circulation of texts bemoaning word processors [Warner 2002]. This counterpublic persisted well into the 2000s but only made headway amongst niche users.

This plain-text focused counterpublic mutated in response to the neoliberalization of higher ed. Rather than viewing plain-text as “sustainable scholarship,” academic lifehacking and minimal computing embraced what David Golumbia calls “computationalism,” an ideological construct postulating social, political, economic, and cultural ills as cybernetic systems to be optimized — or “hacked.” Minimal computing’s fetishization and casting of plain-text as transformative mirrors similar discourses amongst older lifehacking enthusiasts. Although plain-text scholarship is not representative of minimal computing as a totality, minimal computing's emphasis on workflows leads to few of the supposed benefits advocates profess, and in many cases, worsens inequalities.

Daring Markdown

In 2004, John Gruber, a user interface designer and blogger, released a new markup language, entitled Markdown, featuring what he claimed was simple and self-evident syntax. “The best way to get a feel for Markdown's formatting syntax,” he asserted, “is simply to look at a Markdown-formatted document” [Gruber 2004]. Gruber’s language relied on a set of simple conventions mimicking those in email correspondence. Users could surround words in single and double asterisks to signal italics and bold fonts, respectively, and create header levels by preceding header titles with a corresponding number of hashtags (i.e., # for H1 and ## for H2). This simplicity was an explicit design goal that permeated Markdown's design language and resulted in its quick adoption. 1

Markdown's simplicity arises from its composition and plain text file format. “The overriding design goal of Markdown,” writes Gruber, “is the idea that a Markdown-formatted document should be publishable as-is, *as plain text*, without looking like it's been marked up with tags or formatting instructions” [Gruber 2004, emphasis added]. According to the Unicode Standard, plain text files are a “pure sequence of characters” that “contain enough information to permit the text to be rendered legibly, and nothing more” [Allen et al. 2012, 18]. It contrasts plain text with styled or rich text, which is “any text representation consisting of plain text plus added information such as a language identifier, font size, color, hypertext links, and so on” [Allen et al. 2012, 18]. Plain text files benefit from simplicity, portability across different platforms, and longevity. These qualities make it more than just a file type. For Dennis Yi Tenen, plain text is an “editorial method of text transcription that is both 'faithful to the text of its source' and is 'easier to read than the original 2

documents'" [Tenen 2017, 3]. In contrast to the maximization of "system-centric ideals," he sees plain text as embodying minimalism [Tenen 2017, 3].

Plain text exposes the structure of digital encoding and resists what N. Katherine Hayles calls the "flickering signifier." In Saussurean semiotics, signifiers are unitary, durable, and lack internal structures. In contrast, Hayles argues that "flickering signifiers" infuse computational systems [Hayles 1999, 25–49]. These are signs embedded in layered informatics where small actions, like the press of a key, can cause massive changes in hardware and software. They literally "flicker" due to computer monitors refreshing their presence multiple times per second, resulting in observers always witnessing them in a state of flux [Gitelman 2002]. While plain text relies on the same computational hardware, it provides a voyeuristic look into what Tenen calls the "textual lamination" of digital materials [Tenen 2017, 145–156]. This textual lamination refers to the ways that the digital binds text on a screen in a series of physical and networked computational infrastructures. 3

Markup languages, like Markdown, also allow writers to structure plain-text documents without aesthetic differentiations in typography. According to the Unicode Standard, markup languages function by "interspersing plain text data with sequences of characters that represent the additional data structure" [Allen et al. 2012, 19]. By preserving sequences of characters as plain text, these languages can separate tags from "real content." For many advocates, this mechanism allows writers to focus on content while leaving design issues to others. 4

The separation of form and content comes with challenges. Chief amongst these is the fragmentation of language protocols as practitioners expand capabilities. Gruber designed Markdown to have a limited feature set, but his decisions bumped up against practicality as popular websites such as GitHub, Reddit, Stack Exchange, Tumblr, and WordPress began permitting users to post in Markdown. New Markdown parsers and variations quickly emerged [Team Vivaldi 2018]. For Gruber, these "flavors" provided evidence of open ingenuity rather than chaos. When asked why he did not seek standardization, Gruber responded "I believe Markdown's success is *due to*, not in spite of, its lack of standardization. And its success is not disputable" [Gruber 2014]. 5

During the flurry of activity to expand Markdown, John MacFarlane, Professor of Philosophy at UC Berkeley, released pandoc, a command-line utility that reads various Markdown flavors and outputs them to different formats [MacFarlane 2020]. This allowed Markdown to serve as a protocol for creating diagrams, slide decks, and even websites. Most importantly, pandoc provided a way for Markdown to include citations and footnotes that resulted in it gaining popularity amongst a niche group of academics for scholarly research. 6

Pandoc permitted Markdown to serve as a viable option for various genres of writing, but what does Markdown have to do with minimal computing? According to Alex Gil, "minimal computing is the application of minimalist principles to computing" [Gil 2015]. He draws inspiration from designer and architect Ernesto Oroza's extemporal ethnography of Cuba's DIY culture. In Havana, Oroza finds a city where participants resist bureaucratic intervention and create idiosyncratic habitats. He stresses the "moral modulor," who by necessity sees the city through a lens of survival [Oroza 2020]. "The moral modulor is an individual who has the impulse to rebuild human life, and this is something he does for his children or his family," states Oroza. "His condition allows him to discriminate against the superfluous or the useless" [Oroza 2016]. Although these practices in Havana may seem far from digital humanities research, they provide a paradigm for how digital humanities scholars and practitioners should view infrastructure more broadly. Put another way, Gil sees in Havana's "architecture of necessity" a catalyst for a digital humanities community that discards — read: "minimizes" — the extraneous. He asks us to look self-reflexively at our operations and orient them around the question of "What do we need?" [Gil 2015]. 7

Minimal computing includes a myriad of definitions and practices. According to Jentery Sayers, the "minimal" in minimal computing can refer to design, usage, consumption, maintenance, barriers, Internet, externals, space, and technical language. Yet, he leaves open the possibility for other forms of minimalism: "I did not account for all the minimals or their particulars here" [Sayers 2016]. Likewise, Gil states, "minimal computing is in the eye of the beholder" [Gil 2015]. 8

Due to the often-ambiguous definitions of minimal computing, we must turn to self-identified practitioners to understand its relationship with Markdown. As Gregory Bateson notes, cultural activities, such as play, serve as meta- 9

communication through their self-referential nature [Bateson 2008]. In minimal computing, plain-text scholarship in Markdown signals, “This is minimal computing.” This switch from the ontological to the methodological is common when defining the digital humanities. As Rafael Alvarado notes, digital humanities, like minimal computing, has no real definition, and it is better to see digital humanities as a social field rather than an ontological one:

> Instead of a definition, we have a genealogy, a network of family resemblances among provisional schools of thought, methodological interests, and preferred tools, a history of people who have chosen to call themselves digital humanists and who in the process of trying to define the term are creating that definition. [Alvarado 2011]

10 As we look for a “genealogy” of minimal computing tools, Markdown emerges as a recurring theme. Jentery Sayers and Alex Gil, self-identified minimal computing practitioners, frequently tout Markdown-inspired static site generators and plain-text scholarship as epitomizing minimal values and assuring sustainability. For example, in an analysis of Tenen and Grant Wythoff's workflow for sustainable plain-text scholarship in Markdown, Gil praises the workflow’s ability to produce “minimal knowledge with the production of a minimal artifact, without creating necessary friction for the readers” [Gil 2015]. He laments “user-friendly” interfaces for their potential to lead to what Matthew Kirschenbaum calls a “haptic fallacy”: “the belief that electronic objects are immaterial simply because we cannot reach out and touch them” [Kirschenbaum 2003]. Likewise, Sayers asserts that Jekyll, a Markdown reliant static site generator, is representative of “minimal design” [Sayers 2016]. Jekyll forms the basis for prominent minimal computing websites, such as those for the Minimal Computing Working Group, and projects, such as Ed. and Wax. We should note that Markdown is, of course, not the totality of minimal computing practices.

11 Markdown as a digital humanities practice is noticeable for not just its technical innovation but the discourses surrounding it. These discourses stress Markdown as a force for sustainable scholarship, easing barriers of entry to digital humanities in the Global South, and as a means for subverting governmental surveillance. According to Bradley Dilger and Jeff Rice, “Despite the predominant roles markup plays in online writing, and despite the social, cultural, rhetorical, and technological implications of these roles, markup is often taken for granted as merely the code behind the text” [Dilger and Rice 2010]. In response, this article explores the sustainability of Markdown as a digital humanities practice by looking back to a key precursor: Donald Knuth’s TeX.

12 Readers may wonder why I focus on TeX and not other markup languages, such as HTML or TEI. Markup languages emerged in the 1960s when users used markups or “flags” to distinguish between lower and upper-case letters. These flags were often proprietary, making it difficult for different computational platforms to interact with one another. In 1967, Michael Kay implored scholars to create a “standard code in which any text received from an outside source can be assumed to be” [Kay 1967]. The most influential of these markup languages was the Generalized Markup Language (GML) created by Charles Goldfarb, Edward Mosher, and Raymond Lorie at IBM in 1969. GML served as a foundation for the Standard Generalized Markup Language (SGML) released in 1986, which in turn provided a protocol for creating new markup languages like HTML and XML [Goldfarb 1999].

13 TeX and Markdown are notable for their embrace by scholars as computational tools for printed text. This is not to say that they do not use other markup languages, such as HTML and TEI, to disseminate research. In fact, digital humanities scholars often stress alternative publishing platforms, such as online exhibits, blogs, and digital maps as meaningful forms of scholarship themselves. However, this is in contrast to TeX and Markdown where the goal is to create a more efficient workflow to output digital materials in analog formats like academic journals and monographs.

14 In this paper, TeX's development and community provide a case study and catalyst for the social and cultural development of Markdown as a minimal computing practice. From the 1970s to the 1990s, academic researchers used TeX to construct plain-text scholarship in mathematics and the hard sciences to enhance typographical output. Most academics saw these concerns as holding marginal importance and abandoned TeX for What You See Is What You Get (WYSIWYG) word processors, especially in the humanities. Drawing on queer theorist Michael Warner, I argue that the community surrounding TeX responded reactionarily to these transformations by forming a counterpublic constituted through a myriad of texts that bemoaned word processors [Warner 2002]. This counterpublic persisted well into the

2000s but only made headway amongst a niche scholarly audience. After exploring TeX, I conclude by looking at the lessons it provides digital humanities scholars about embracing minimal computing practices like Markdown-based scholarship. As I make evident, although minimal computing encompasses a range of modalities, discourses surrounding Markdown mirroring those of TeX should give pause to those seeking to implement it in practice.

Taking TeX Seriously

15 To understand Markdown-based scholarship in the digital humanities and growing advocacy for its use, I turn to the history of computing. The history of computing remains a niche field and has had little overlap with the digital humanities. Historians of computing have dismissed the latter as narrowly focusing on career development and sidelining issues that concern other humanities scholars. As Thomas Haigh — echoing Bruno Latour's criticism of modernity concludes — “We have never been digital” [Haigh 2006]. Yet, computing's history can provide valuable insights into minimal computing practices.

16 Given markup's popularity amongst programmers and technical writers, it may be tempting to see it as an extension of programming. However, the catalyst for plain-text scholarship stems from developments and concerns about digital typography best exemplified by Donald Knuth’s TeX. A mathematical prodigy, Knuth won a scholarship to Case Western Institute, where he tinkered with the university’s IBM 650, an early mainframe computer, in his spare time. After switching concentrations from physics to mathematics, Knuth graduated with a Bachelor of Science in mathematics and, by a special vote of the faculty, a Master of Science for exceptional work. He went on to earn a doctorate in mathematics from the California Institute of Technology and joined its faculty as an Assistant Professor of Mathematics in 1963 [Walden 2019].

17 Knuth began writing his magnum opus, *The Art of Computer Programming*, originally a comprehensive work about compilers, full time in 1962 but took a break to investigate the statistical properties of linear probing. This break had a profound impact on Knuth who proposed a broader project to his publisher about computational algorithms. By the summer of June 1965, he had completed three thousand handwritten pages (roughly two thousand pages of printed text), and Addison-Wesley agreed to publish the revised work in seven parts. With Knuth’s desire for *The Art of Programming* to mirror the totality of computer science algorithms, he continued to amend the work throughout his career. While revising the second edition of the book in 1976, Knuth received galley proofs from his publisher. Addison-Wesley had adopted a new digital-infused workflow, and Knuth became so aghast at the typographical quality that he threatened to quit the project altogether [Knuth 2007].

18 Pausing from writing *The Art of Programming*, Knuth sought to explore how computer science could enhance typography after seeing a high-resolution digital typesetting machine in 1977. By dividing a page into discrete sections (pixels), he searched algorithmically for the ideal place for ink (1) or not (0). “As a computer scientist, I really identify with patterns of 0’s and 1’s; I ought to be able to do something about this,” noted Knuth [TeX Users Group 2021].

19 Knuth's work on typography formed the basis for TeX, and he dedicated his 1977–1978 academic sabbatical to the project. Approximately a year later, the American Mathematical Society invited Knuth to give a lecture that he used to stress TeX’s superiority and mathematical underpinning. For the attending academic audience, TeX offered numerous benefits. As the TeX User Group’s official history notes, creating academic articles at the time was expensive and relied on proprietary software. A document created in one software program often rendered differently on competing systems. TeX solved these issues by providing a free markup system that was portable across platforms and geared towards academic work [TeX Users Group 2021].

20 We should pause to underline that TeX's development had little to do with easing composition as later enthusiasts would argue. While the portability of TeX did allow researchers to exchange files without vendor lock-in, researchers were more concerned about preserving aesthetics rather than stripping away content and form. They exchanged TeX files with typographical information and choices affixed in document front matter so that others could compose facsimiles rather than leave questions of appearance to them.

21 Despite TeX’s advantages, it had a steep learning curve. Seeking to assist newcomers, computer scientist Leslie

Lamport assembled a series of macros for TeX, which he called LaTeX, in 1984. LaTeX took cues from Scribe, a markup language Brian Reid created for his doctoral dissertation that popularized “logical design,” a philosophy that separated the production of content from appearance. Reid claimed that Scribe allowed writers to focus on their writing rather than aesthetics, and by utilizing a small set of declarations, they could still output their creations to different formats [Reid 1981]. Despite its innovations, many remained apprehensive about what they saw as Scribe’s rigidity. Early responses touted LaTeX as “Scribe liberated from inflexible formatting control” and “TeX for the masses” [Mittelbach et al. 2004]. In his original manual *LaTeX: A Document Preparation System*, Lamport acknowledges his desire for a simpler system. “In turning TeX into LaTeX, I have tried to convert a highly-tuned racing car into a comfortable family sedan,” writes Lamport, “The family sedan isn't meant to go as fast as a racing car or be as exciting to drive, but it's comfortable and gets you to the grocery store with no fuss” [Lamport 1986, xiii].

The rhetoric around LaTeX differed significantly from TeX. Lamport, like Reid, saw logical design as a chance to ease composition and to prevent vendor lock-in. This is not to say that concerns of typography disappeared, but they had diminished considerably. TeX’s typographic benefits originally drew Lamport’s interest, but his assemblage of easy-to-use macros shifted how advocates pitched TeX's benefits and would serve as a catalyst for later claims by Markdown enthusiasts. Still, most academics continued to find markup-oriented scholarship to be unnecessarily difficult to use, and with WYSIWYG word processing software more readily available, all but the most diehard users defected, especially in the humanities. 22

Word processing had emerged during the 1970s as an organizational paradigm for centralizing corporate information [Haigh 2006]. Businesses created word processing departments that focused on text production in an attempt to ease the burden on other departments. Manufacturers, such as IBM and Wang, created standalone systems for this new information management paradigm, and by the early 1980s, these machines were commonplace in corporate America. Word processing software dedicated solely to text editing, on the other hand, was of marginal purpose. As Thomas Haigh notes: 23

> It … seemed no more sensible to use a computer to edit than to travel to the shopping mall in a supersonic fighter jet. Only the plummeting cost of interactive computing could turn an absurd luxury into an expensive tool with economic justifications in specialized fields, and eventually into an inexpensive office commonplace. [Haigh 2006] 24

As personal computers expanded into business environments during the late 1980s, word processing's meaning shifted to refer to standalone software for manipulating text. WYSIWYG editors provided real-time rendering of how text would look on paper if the user printed it. Various manufacturers created competing programs throughout the decade, but Microsoft Word emerged as the market leader by the middle of the 1990s, in turn becoming the de facto standard for academic manuscripts. 25

If TeX remained a novel typographical system, its impact on the academic community would be minimal as many would have abandoned it when WYSIWYG editors became more efficient at displaying mathematical formulas. However, discourses about TeX's utility shifted with the widespread adoption of personal computing. Rather than emphasizing its technical features, advocates posited it as resisting word processing’s dominance as a compositional tool. Their discourses focused on two key trends. First, critics asserted that word processing software's proprietary formats threatened open access of scholarly information. Second, they saw word processors as “distracting” during intellectual labor. Matthew Kirschenbaum notes, “Word processing … shapes and informs literary subjects — the persons who inhabit the system (and economy) of literature, green-screeners or otherwise” [Kirschenbaum 2016]. As writers change their tools, these tools shape their composition and orient them into a techno-social framework centered around the process of writing. It is in this social milieu that TeX became a social force for connecting detractors of word processing, rather than just a typographical apparatus. 26

By the late 1990s and early 2000s, TeX’s users formed a counterpublic united through texts lamenting word processors. As Michael Warner demonstrates, counterpublics shape queer identity and challenge Jürgen Habermas' “bourgeois” public sphere [Warner 2002]. Habermas correlates the public sphere's rise in the eighteenth century with the emergence 27

of coffeehouses, cafes, salons, and reading clubs in France, England, and Germany. In these spaces, the notion of citizenship blended with an abstracted universal body [Habermas 2015]. Nancy Fraser criticizes this universalizing impetus of the public sphere and draws attention to “subaltern counterpublics” that emphasized identity and power [Fraser 1990].

28 Michael Warner draws on Fraser’s conception to make evident how texts' circulations are critical in forming counterpublics. He notes that the “public” refers to a myriad of contradictory definitions. On one hand, *the public* refers to a totality whose members are defined through a set of universalities. *A public*, on the other hand, refers to a subset within this broader public. Warner considers a third definition of public. “This kind of public,” notes Warner, “comes into being only in relation to texts and their circulation. It exists *by virtue of being addressed*” [Warner 2002, original emphasis]. The audience's attention to the text creates a set of relationships, and in doing so, formulates a distinct “poetic” worldview [Warner 2002].

29 As corporations embraced word processing during the 1990s, desktop publishing constituted a specific public defined through the circulation of texts, advertisements, images, and videos addressing a neoliberal workforce. According to Jamie Peck and Adam Tickell, neoliberalism has two distinct phases of state mobilization: a “roll-back” and a “roll-out.” In the roll-back period of the 1980s, political elites mobilized the state to create mass deregulation and implemented supply-side economic policies. These policies catapulted Margaret Thatcher and Ronald Reagan into political office but reached their political limit by the 1990s. This led to neoliberalism's “roll-out,” where policymakers mobilized the state to regulate and discipline the same individuals dispossessed by neoliberalization, namely African American and Latino communities [Peck and Tickell 2002]. Eschewing collective political action, neoliberalism imparted ideas of self-entrepreneurship, confidence, and prudence onto political subjects.

30 Neoliberal market ideology, as Fred Turner shows, intersected with advocacy for deregulation by cybercultural entrepreneurs [Turner 2006, 175–206]. For neoliberal policymakers, computers with WYSIWYG desktop publishing software were essential to launching micro-marketing campaigns, assessing the labor force quantitatively, and accelerating policy report and white paper creation. In this environment, TeX served as a counterpublic against capitalist exploitation of intellectual labor. Although most TeX users were white, their outlook on academic publishing pushed them against the neoliberalizing trend of efficiency and commercial concerns. As historians have noted about the formation of this community:

> 31 TeX resulted in a worldwide community of users, developers, and user groups evolved, largely disconnected from the more conventional desktop publishing world driven by commercial concerns of publishers and desktop publishing system vendors (so very different than Knuth’s concerns for TeX). This community remains vibrant today, 40 years later, and is an important branch in the development of desktop publishing. [Beeton, Berry, and Walden 2018]

32 The concentrated community backlash to business practices was in part due to Knuth’s curation of enthusiasts around his software. On February 22, 1980, a small group of enthusiasts bound through Knuth's influence formed the TeX Users Group (TUG). According to Beeton, Berry, and Walden, “The formation of TUG was an important step in TeX's becoming widely popular and in TEX development activity eventually becoming independent of Stanford” [Beeton, Berry, and Walden 2018]. From 1982, Knuth also began to hold a “MetaFont for Lunch Brunch” to discuss typography — the name alludes to his complementary technology for specifying fonts, which failed to gain traction [Beeton, Berry, and Walden 2018]. The group and Knuth’s academic connections ensured that a series of graduate researchers, faculty, and outside contributors would continue to develop and notably, write about the system. By the decade's end, TeX User Groups had close to four thousand members and growing international reach. They began publishing a journal entitled *TUGBoat* with support from the American Mathematical Society, and the circulation of this new journal helped solidify the counterpublic.

33 As mentioned earlier, concerns about word processing's growing influence in the academic environment and distracting tendencies dominated texts about TeX. For instance, an early critique of word processors underlies Lamport's discussion of LaTeX. In his original manual, he compares logical design to visual design (another term for “what you see

is what you get"). According to Lamport, LaTeX "encourages better writing" by causing writers to focus on their document's logical structure [Lamport 1986, 8]. It also allows for better focus on text's composition rather than design: "LaTeX was designed to free you from formatting concerns, allowing you to concentrate on writing. If, while writing, you spend a lot of time worrying about form, you are probably misusing LaTeX" [Lamport 1986, 8]. This critique became explicit in Lamport's 1994 revised manual. He notes that when he wrote LaTeX, there were few facilities for authors to typeset their documents whereas they had become commonplace by the mid-1990s. He goes on to explicitly lambast the development of "WYSIWYG programs [that] replace LaTeX's logical design with visual design" [Lamport 1986, 7].

34 Later guides for LaTeX — which became the de facto flavor of TeX — would reassert claims about word processing's perceived flaws for academic work. In 1989, for instance, Jon Warbrick released a condensed version of Lamport's manual. In it, he writes, "A visual system makes it easier to create visual effects rather than a coherent structure; logical design encourages you to concentrate on your writing and makes it harder to use formatting as a substitute for good writing" [Warbrick 1988, 2]. Another widely circulated manual by Gavin Maltby in 1992, entitled *An Introduction to TeX and Friends*, notes, "Essential to the spirit of TeX is that *it formats the document whilst you just take care of the content*, making for increased productivity" [Maltby 1992, 3, original emphasis]. Another self-described "polemical rant in favor of TeX as opposed to word processors" by Allin Cottrell, Professor of Economics at Wake Forest University, describes Knuth's invention as a panacea to WYSIWYG word processors' faults. "The word processor is a stupid and grossly inefficient tool for preparing text for communication with others," contends Cottrell [Cottrell 1999].

35 The conflation of TeX with logical design focused on content creation rather than typography continues into the present. For example, the website for the LaTeX project prominently states the importance of separating content from presentation: "LaTeX is not a word processor! Instead, LaTeX encourages authors not to worry too much about the appearance of their documents but to concentrate on getting the right content " [LaTeX Project 2020]. It contrasts LaTeX with how an author would format a document by determining layout and font. "This has two results," notes the site, "authors wasting their time with designs; and a lot of badly designed documents!" [LaTeX Project 2020]

36 In short, by the late 1990s and early 2000s, a counterpublic emerged through blogs, newspaper articles, and online discussion forums, positioning TeX as an alternative to word processing for academic production. As Warner makes evident, "texts themselves do not create publics [or counterpublics], but the concatenation of texts through time. Only when a previously existing discourse can be supposed, and a responding discourse can be postulated, can a text address a public" [Warner 2002].

37 Readers in mathematical fields may wonder how TeX can be a counterpublic given its popularity in their disciplines. It is important to emphasize that counterpublics often gain wide support amongst certain subgroups. "A counterpublic maintains at some level, conscious or not, an awareness of its subordinate status," argues Warner. "The cultural horizon against which it marks itself off is not just a general or wider public but a dominant one," he continues [Warner 2002]. For instance, queer communities may have consumer literature challenging heteronormative relationships, but these works remain niche amongst mainstream publishers. Likewise, even though TeX has adoption in mathematical fields, it has little support in the corporate world, governmental sector, or most academic disciplines.

The Pitfalls of Markdown

38 TeX's origins and subsequent development provide unique insight for scholars of minimal computing. As mentioned earlier, minimal computing advocates often stress Markdown and Markdown-adjacent tools as representative, although not the totality of, their practice. In this brief conclusion, I assess four cautious lessons that TeX provides minimal computing practitioners about the adoption of Markdown-based scholarship in the digital humanities.

39 The first lesson TeX provides is that the adoption of niche tools rarely eases barriers of entry and access. With the rise of WYSIWYG word processors, digital composition opened up to a wider range of individuals. Works that were in TeX pushed against this mainstream adoption. As a result, with growing familiarity of computers throughout the 1990s, only those that knew what TeX was, found its system easier to manage, and most importantly, had the specialized training often gained in graduate school to manipulate it were able to contribute. Anne B. McGrail draws attention to a similar issue in minimal computing in reference to her community college students:

> My students might read Gil's and Sayer's pieces but they will likely write about them using Google or Word on their laptops and email me using Outlook — transgressing minimal computing principles at every keystroke. For these definitions of minimal computing themselves point to a set of assumptions for humanistic computing that are worth considering in CC students' digital lives. [McGrail 2017]

40 Second, TeX shows that plain-text scholarship, like that written in Markdown, requires a large infrastructure to support it and is far from minimal. In fact, criticisms of large-scale infrastructures as limiting forces in the digital humanities well precede the Minimal Computing Working Group. In a 2012 study of digital humanities projects, Joris van Zundert concludes:

41 > At least as far as digital humanities research is concerned, there is little benefit to be expected from the current large infrastructure projects. Their all-purpose nature enforces a generalized strategy aimed at the establishment of standards which is at odds with innovative, explorative research. Being standards-driven, institutionally bound, and at worst enforcing specific implementations, they are platforms of exclusiveness. [Van Zundert 2012]

42 Echoing Gil's later remark that minimal computing advocates should orient themselves around the question of "what we need," Zundert writes that digital humanities infrastructures "should indeed be the simplest thing that could possibly work" [Van Zundert 2012]. Yet, TeX shows that most supporters, including Knuth, were often only able to contribute to the project due to the support of their respective academic institutions. Likewise, Markdown-oriented site generators, such as Jekyll, stress the "free" hosting of GitHub, a company Microsoft bought for over $7.5 billion and backs with its large corporate infrastructure. The impact of Microsoft's large server network, muddled with bit rot and electrical waste, is rarely taken into account.

43 Third, TeX shows that plain-text scholarship is not significantly more future-proof or sustainable compared to more proprietary alternatives. In regards to their workflow, Tenen and Wythoff argue, "Writing in plain text guarantees that your files will remain readable ten, fifteen, twenty years from now." They contrast this with proprietary formats, like Word or Google Docs, that "may go the way of WordPerfect in the future" [Tenen and Wythoff 2014]. While the "plain sequence of characters" is likely to be readable, the scripts scholars rely on to convert their documents into more acceptable publishing standards have limited upkeep. As TeX grew in popularity, support for older versions waned leaving many scholars without an easy means of reconverting their documents. To turn back Markdown, it's important to remember that a key reason for its growing popularity is its ability to transform into a wide variety of formats through pandoc. Although a variety of coders contribute to it, ultimately, it relies on the leadership of MacFarlane, and it is unclear how long it would survive without his input. In contrast, large organizations and governmental agencies that have adopted proprietary software along with the corporations that produce this software have a much greater incentive to maintain compatibility.

44 Finally, plain-text scholarship provides little security for vulnerable political activists as claimed by boosters. Gil notes how sites off-grid can assist political revolution. He highlights a project entitled No Connect that he and his colleagues developed at Columbia University's Group for Experimental Methods in the Humanities [Gil and Tenen 2015]. No Connect is a theme for Jekyll meant for users without an internet connection. Presumably, this allows political dissidents to distribute subversive and sensitive materials through Sneakernets, informal networks for transferring information through physical mediums such as CD-ROMs, floppy disks, and USB drives. While such sites may circumnavigate traditional computer networks, they come with their challenges. For instance, governmental agencies posing as bureaucrats can steal or copy them. These same individuals may produce malware and keyloggers that are automatically installed when a user places the USB drive in their computer, compromising more elements of the Sneakernets. A popular exploit places a large surge of electricity through the USB port frying the machine [Angelopoulou et al. 2021]. Later developments of TeX/LaTeX introduced security risks due to their reliance on macros [Checkoway, Shacham, and Rescorla 2010]. While Markdown does not rely on as many macros, its convoluted conversion structure leaves it open to similar vulnerabilities.

45 Of course, minimal computing is divergent and growing, and the emphasis on plain-text scholarship that this article

highlights does not represent the field's totality. Many advocates of minimal computing have little focus on markup languages. Nonetheless, by historicizing and contextualizing Markdown through my analysis of TeX, I have sought to show some of the pitfalls that may occur through widespread adoption. In the future, these issues may be solved, but TeX's lessons should give digital humanities scholars a cautionary warning.

Works Cited

Allen et al. 2012 Allen, Julie D. et al. "The Unicode Standard." *Unicode Consortium,* 2012. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.683.9133&rep=rep1&type=pdf.

Alvarado 2011 Alvarado, Rafael. "The Digital Humanities Situation." *The Transducer*, May 11, 2011. http://transducer.ontoligent.com/?p.

Angelopoulou et al. 2021 Angelopoulou, Olga, Seyedali Pourmoafi, Andrew Jones and Garauv Sharma. "Killing Your Device via Your USB Port," 2020. http://wrap.warwick.ac.uk/137908/1/WRAP-killing-your-device-via-USB-port-Angelopoulou-2020.pdf.

Bateson 2008 Bateson, Gregory. *Steps to an Ecology of Mind,* Chicago; London: University of Chicago Press, 2008.

Beeton, Berry, and Walden 2018 Beeton, Barbara, Karl Berry, and David Walden. "TEX: A Branch in Desktop Publishing Evolution, Part 1." *IEEE Annals of the History of Computing* vol. 40.3 (2018): 78–93.

Checkoway, Shacham, and Rescorla 2010 Checkoway, Stephen, Hovav Shacham and Eric Rescorla. "Are Text-Only Data Formats Safe? Or, Use This LATEX Class File to Pwn Your Computer," 2010. https://hovav.net/ucsd/dist/texhack.pdf.

Cottrell 1999 Cottrell, Allin. "Word Processors: Stupid and Inefficient," June 29, 1999. http://ricardo.ecn.wfu.edu/~cottrell/wp.html.

Dilger and Rice 2010 Dilger, Bradley J. and Jeff Rice. "Making a Vocabulary for <HTML>." In *A to <A>: Keywords of Markup,* edited by Bradley J. Dilger and Jeff Rice, xi–xxiv. Minneapolis, MN: University of Minnesota Press, 2010.

Fraser 1990 Fraser, Nancy. "Rethinking the Public Sphere: A Contribution to the Critique of Actually Existing Democracy." *Social Text* 25/26 (1990): 56–80.

Gil 2015 Gil, Alex. "The User, the Learner and the Machines We Make." *Minimal Computing: A Working Group of GO::DH*, May 21, 2015. https://go-dh.github.io/mincomp/thoughts/2015/05/21/user-vs-learner/.

Gil and Tenen 2015 Gil, Alex and Dennis Yi Tenen. "No Connect." *Group for Experimental Methods in the Humanities,* July 24, 2015. https://xpmethod.columbia.edu/knowledge-design-studio/2015-07-24-no-connect.html.

Gitelman 2002 Gitelman, Lisa. "Materiality Has Always Been in Play: An Interview with N. Katherine Hayles." *Iowa Journal of Cultural Studies* vol. 2.1 (2002): 7–12.

Goldfarb 1999 Goldfarb, Charles F. "The Roots of SGML: A Personal Recollection." *Technical Communication* vol. 46.1 (1999): 75-83.

Gruber 2004 Gruber, John. "Markdown." *Daring Fireball*, December 17, 2004. https://daringfireball.net/projects/markdown/.

Gruber 2014 Gruber, John. "Twitter /@gruber: @comex I believe Markdown's success is *due to*, not in spite of, its lack of standardization. And its success is not disputable." September 3, 2014, 11:09 PM. https://twitter.com/gruber/status/507364924340060160.

Habermas 2015 Habermas, Jürgen. *The Structural Transformation of the Public Sphere.* Cambridge: Polity, 2015.

Haigh 2006 Haigh, Thomas. "Remembering the Office of the Future: The Origins of Word Processing and Office Automation." *IEEE Annals of the History of Computing* vol. 28. 4 (October 2006): 6–31. https://doi.org/10.1109/MAHC.2006.70.

Hayles 1999 Hayles, Katherine. *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics.* Chicago, IL: University of Chicago Press, 1999.

Kay 1967 Kay, Martin. "Standards for Encoding Data in a Natural Language." *Computers and the Humanities* vol. 1.5 (1967): 170–77.

Kirschenbaum 2003 Kirschenbaum, Matthew G. "Virtuality and VRML: Software Studies after Manovich." In *The Politics*

of Information. Stanford, CA: Alt-X Press, 2003. https://electronicbookreview.com/essay/virtuality-and-vrml-software-studies-after-manovich/.

Kirschenbaum 2016 Kirschenbaum, Matthew G. *Track Changes: A Literary History of Word Processing*. Cambridge, MA: Harvard University Press, 2016. https://doi.org/10.4159/9780674969469.

Knuth 2007 Knuth, Donald. "Oral History." Interview by Edward Feigenbaum. *Computer History Museum*, March 14, 2007. https://www.computerhistory.org/collections/catalog/102658053.

LaTeX Project 2020 LaTeX Project. "An Introduction to LaTeX." *LaTeX Project*. Accessed August 19, 2020. https://www.latex-project.org/about/.

Lamport 1986 Lamport, Leslie. *LaTeX: A Document Preparation System: User's Guide and Reference Manual.* Reading, MA: Addison-Wesley, 1986.

MacFarlane 2020 MacFarlane, John. "About Pandoc." *Pandoc*, August 20, 2020. https://pandoc.org/.

Maltby 1992 Maltby, Gavin. *An Introduction to Tex and Friends*, 1992. http://faculty.washington.edu/stacy/prb/pdf/intro_to_tex_and_friends.pdf.

McGrail 2017 McGrail, Anne. "Open Source in Open Access Environments: Choices and Necessities." *Minimal Computing: A Working Group of GO::DH*, February 17, 2017. https://go-dh.github.io/mincomp/thoughts/2017/02/17/mcgrail-choices/.

Mittelbach et al. 2004 Mittelbach, Frank et al. *The LaTeX Companion*. Boston, MA: Addison-Wesley Professional, 2004.

Oroza 2016 Oroza, Ernesto. "Interview with Ernesto Oroza." Interview by Alex Gil. In *Debates in the Digital Humanities* 2016, edited by Matthew K. Gold and Lauren F. Klein, 184-93. Minneapolis: University of Minnesota Press, 2016.

Oroza 2020 Oroza, Ernesto. "About." *Architecture of Necessity*, August 20, 2020. http://architectureofnecessity.com/about/.

Peck and Tickell 2002 Peck, Jamie and Adam Tickell. "Neoliberalizing Space." *Antipode* vol. 34.3 (2002): 380–404.

Reid 1981 Reid, Brian Keith. "Scribe: A Document Specification Language and Its Compiler." Ph.D. thesis. Carnegie Mellon University, 1981. https://search.proquest.com/docview/303093737/abstract/BD9D8F366D5D4038PQ/1.

Sayers 2016 Sayers, Jentery. "Minimal Definitions-Minimal Computing." *Minimal Computing: A Working Group of GO::DH*, October 2, 2016. http://go-dh.github.io/mincomp/thoughts/2016/10/02/minimal-definitions/.

TeX Users Group 2021 TeX Users Group. "History of TeX - TeX Users Group." *TeX Users Group*. Accessed October 22, 2021. https://www.tug.org/whatis.html.

Team Vivaldi 2018 Team Vivaldi. "What is Markdown and Why Should You Care." *Vivaldi Browser*, November 13, 2018. https://vivaldi.com/blog/markdown-in-vivaldi-notes/.

Tenen 2017 Tenen, Dennis. *Plain Text: The Poetics of Computation*. Stanford, CA: Stanford University Press, 2017.

Tenen and Wythoff 2014 Tenen, Dennis, and Grant Wythoff. "Sustainable Authorship in Plain Text Using Pandoc and Markdown." *Programming Historian* 3, March 19, 2014. https://programminghistorian.org/en/lessons/sustainable-authorship-in-plain-text-using-pandoc-and-markdown.

Turner 2006 Turner, Fred. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. Chicago, IL: University of Chicago Press, 2006.

Van Zundert 2012 Van Zundert, Joris. "If You Build It, Will We Come? Large Scale Digital Infrastructures as a Dead End for Digital Humanities." *Historical Social Research/Historische Sozialforschung* (2012): 165–86.

Walden 2019 Walden, David. "Donald E. Knuth - A.M. Turing Award Laureate." *Association for Computing Machinery*, 2019. https://amturing.acm.org/award_winners/knuth_1013846.cfm.

Warbrick 1988 Warbrick, Jon. "Essential LaTEX," 1988. Accessed March 22, 2022. http://tug.ctan.org/info/latex-essential/ess2e.pdf.

Warner 2002 Warner, Michael. "Publics and Counterpublics." *Public Culture* vol. 14.1 (2002): 49–90.