Scholarship in interaction: case studies at the intersection of codework and textual scholarship

Zundert, J.J. van

SCHOLARSHIP IN INTERACTION

CASE STUDIES AT THE INTERSECTION OF CODEWORK AND TEXTUAL SCHOLARSHIP

JORIS J. VAN ZUNDERT

Scholarship in Interaction

Scholarship in Interaction

Case Studies at the Intersection of Codework
and Textual Scholarship

Joris J. van Zundert

Scholarship in Interaction

Published by Fox and Badger Publications

ISBN 978 90 90 36460 5

Scholarship in Interaction

Case Studies at the Intersection of Codework
and Textual Scholarship

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op dinsdag 27 september 2022
klokke 15.00 uur

door

Joris Job van Zundert
geboren te Goirle
in 1972

Promotiecommissie

Promotores

Prof.dr. P.F. Wouters
Prof.dr. K.H. van Dalen-Oskam (Universiteit van Amsterdam)

Overige leden

Prof.dr. S. de Rijcke
Prof.dr. L.R. Waltman
Prof.dr. S. Schreibman (Universiteit Maastricht)
Prof.dr. J. Nyhan (University College London)
Em.prof.dr. M. Thaller (Universität zu Köln)

Contents

Acknowledgements

Clearly I am not in a hurry. There was so much to enjoy, why would I have been in a hurry?

One advantage of the long time I took before starting a PhD project and the equally long time I eventually spent to finish writing a dissertation, is that my academic interests meanwhile span twenty years. The academic work during these years has brought me in contact with so many friendly, interesting and interested, brilliant, and sometimes fascinatingly different people. This is what I am truly thankful for. Unfortunately there will not be enough pages in this dissertation to thank all those who crossed my path for a shorter or longer time, and who contributed to my knowledge and understanding of which much has landed on these pages. Many of you became true friends. If you read this at all and if we have met, it is very probable I count you among that group.

Obviously I also want to thank Paul Wouters, my supervisor, and Karina van Dalen-Oskam, my co-supervisor. Without them this dissertation would never have seen the light of day. I thank both for their inspiring thoughts, knowledgeable conversations, and invaluable hints and criticism that improved my thinking and my work. They know as well as I do that one of my pitfalls is not properly finishing stuff: there is always something else to turn a mind to because it is fascinating, interesting, new, or – that dreaded adjective – urgent. So, I thank them both for their patience and for having relentlessly nudged me in the direction of actually finishing this document as well.

Allow me also to thank Henk Wals, former scientific director of the Huygens Institute, who agreed to support my PhD project and who allowed me to work on it as part of my ongoing work at the institute. I thank Henk's successor Lex Heerma van Voss and René Jongerius (director of operations at the time) for actually freeing up space in my assignments to get important parts done. I thank Peter Doorn for granting me a way into academia at all. And I especially thank all my dear colleagues who, throughout the years, have so generously shared knowledge and experience.

Finally, to Leentje and Aafke, to family and friends, to those present and to those gone, I say thank you for making life worthwhile. Leentje and Aafke explain my job to others as "something with computers". I need to do a better job at explaining, but these matters are not simple. At least let us change that to "something with computers and literature", and we will go from there...

Chapter 1

Preliminary Matters

1.1 Introduction

It might come rather as a surprise to her but Anne Beaulieu is to blame for me writing a dissertation at all and, to an extent, for the way it turned out. This work was inspired by a remark she made during an all hands meeting with the project team I was leading at the time. She was a team member as a delegate for the – now unfortunately dissolved – Virtual Knowledge Studio. The Virtual Knowledge Studio[1] studied how digital technology impacts the humanities and the social sciences. Our project investigated and explored some of the potential of computational approaches for humanities research at the Royal Netherlands Academy of Arts and Sciences (KNAW), and we were thus continuously discussing the merits and deficiencies of such methods. At some point Anne remarked that the computational tools offered until then might simply not be good enough to solve humanities research questions in any significant way. Anne's remark was not some Luddite platitude. What she meant was that the questions that humanities researchers tend to ask are very subtle, heterogeneous, based on abductive reasoning, and focused on specifics and idiosyncrasies. In other words, most humanities questions tend to be hermeneutic, concerned with interpretation and with offering various perspectives on the same issue. This creates a stark contrast to computational approaches that are overwhelmingly still quantitative, often deterministic, and driven by a wish for generic solutions. It was Anne's

[1] https://web.archive.org/web/20120529000713/http://virtualknowledgestudio.nl/

remark that made me start to wonder: how do digital methods in the humanities affect interpretation?

The impact of digital technology with respect to textual scholarship is often regarded from the perspective of representation: how does the digital environment or how do digital mechanisms for research and presentation affect the representation of text? Obviously representational effects of remediation are real and substantial. However, representation is both the result and cause of interpretation. Representation in textual scholarship derives from interpretation when a scholarly editor devises a new edition of a text, and at the same time that new edition is provided in order to provoke renewed interpretation. This – as well as the background study below – leads me to believe that the most profound effect of digital technology on textual scholarship has to do with modes, possibilities, objects, and mechanisms of interpretation. Given that interpretation is the central tenet of humanities and especially textual scholarship methodology, this effect deserves attention. The chapters and case studies in this book all share that theme: they regard the impact of digital technology from the perspective of scholarly interpretation.

A pivotal aspect in this respect is the interplay between affordances and tensions created by the digital remediation of the book and especially by its digital re-materialization. The specific material constraints of the physical book have had a formative effect on scholarly conventions. These conventions in turn determine to a large extent the constraints that establish trust and authority in the inferences based on textual criticism. The rather resistant materiality of the book that was key in the formation of the – to use Foucault's terminology – "episteme" of textual scholarship, has been radically eradicated by software technology and replaced by the infinitely moldable materiality of the digital environment. Textual scholars are thus confronted with a radical liberty of expression and a sheer unlimited knowledge space for representation. A question that subsequently intrigues me is how this different digital materiality and this loss of print publishing constraints shape the act of interpretation in textual scholarship. Rather in contrast to the aspects of remediation the possible effects of digital methodology on hermeneutics remain severely understudied. Because of this lack of attention, these changes – that I believe are real – also remain covert and invisible to most scholars.

Much attention has been called to the effects of software and digitalization on society (cf. Coyne 1995; Capurro 2010; Berry 2014). Almost all human actions and transactions in our modern society at some point involve some form of digital technology. Software and digital technology have an immense and still generally underestimated impact on human behavior and thinking (cf. e.g. Hayles 2012). Often these encounters with digital technology are benign and unobtrusive, such as when we are using a hearing aid (setting aside possible privacy issues). But at their worst the interactions remain intentionally covert, such as when digital technologies are used for large scale surveillance in open societies. The massive impact of digital technologies is being portrayed both as benign and disruptive. Revolutionary promises (Shirky 2010; Courant et al. 2006) compete with dystopian prophecies (Morozov 2013; Kirsch 2014). However, I have little handwaving to add on how society as a whole is threatened by softwarization or on how we became post-digital (Berry 2013). I am interested in more concrete examples of how code, digital objects, and people interact – especially how textual scholarship engages with digitality. Pointing to software or Silicon Valley as dark forces in general is a facile simplification. As for instance Paul Wouters (Wouters 2004) and Christine Borgman (Borgman 2007) have suggested, we need better and more precise case studies of what happens when, for instance, a scholarly community sees an influx of digital methods and techniques. Since it is an under-researched domain within the social study of the humanities, Borgman has specifically called attention to the study of the computerization movement in the humanities. I would add that – although I do not pretend or aim to uncover such large scale influences – understanding how the humanities are transformed through digital technology and methodology is pivotal also in view of trying to understand how societies change through such technology. For the humanities position themselves to a large extent as substantially contributing to reflective perspectives on society and culture (cf. e.g. Small 2013; Terras et al. 2013). Understanding how softwarization affects the humanities is thus important to understand the critical and reflective role of the humanities within a post-digital society.

However, by themselves the humanities are a very large domain, with many subfields using various methodologies, asking different research questions, and applying a richness of theoretical frames. We will have to guard against simplistic abstractions that arise from mistaking the high level view of the digitalization process in the humanities as a neatly motivated and progressive computational evolution. This study therefore prefers case study level research and focuses on the computerization movement in textual scholarship alone. But even that digitalization process involves an intricate ecosystem of programmatic factions, individual pragmatism, computational theorism, institutional interests, and so forth. Depending on one's perspective and position that movement can be experienced as a methodological change that has been unfolding since at least the 1940s (Hockey 2004), a revolutionary paradigmatic shift (Bod 2013c) or a technologic whimsical fad (Fish 2013). Thus with regard to this computerization movement in textual scholarship I need to narrow the focus even more. Following Lev Manovich (2013:8): "Paradoxically, while social scientists, philosophers, cultural critics, and media and new media theorists seem by now to cover all aspects of IT revolution, creating a number of new disciplines such as cyberculture studies, Internet studies, game studies, new media theory, digital culture, and digital humanities, the underlying engine which drives most of these subjects -- software -- has received comparatively little attention." The nature and role of software code and their role in reshaping textual scholarship have indeed been very little studied. My various case studies are indeed especially motivated by the urge to understand how the particular affordances and limitations of code, the makers of code, and textual scholars together shape a different scholarship.

I take a particular interest in the fact that code is a form of text with added executability. The performative and manipulative nature of software code lends code aspects of a deferred agency. As a – possibly far from neutral – influence on scholarly interpretation and authority these aspects of code are still poorly understood, and thus I want to understand them better. Perspectives on text in text theory are manifold: text-as-object, text-as-experience, text-as-process, and so forth. Code in a more practical sense turns text quite literally from text-as-object into text-as-process. Alluding to the ideas of Roland Barthes, the manipulability of code and its performative character may create powerful affordances for readerly and writerly activities. If code appears to be an actor in textual scholarship it may cause a partial or complete shift of labor, authority, responsibility, and accountability associated with particular scholarly activities. These shifts may occur between human actors, but also between human and computational or digital mechanisms. How textual scholarship engages with these performative affordances – or not – is a primary research question for me.

The introduction of new technologies in a scholarly domain leads to a particular offset between conventional skillsets and new skills needed to adequately participate in that domain. Does this mean that with the adoption of digital tools code literacy becomes a requisite skill in textual scholarship? And if so, to what fluency should that skill be developed? Moreover, does the ability to read and write code change the scholarly act of interpretation? What overt or covert effects has the interface on a computer's screen on the process of scholarly text interpretation? Are possible hermeneutic effects negligible and may textual scholars simply ignore code's specific semiotics and scholarly performance? Can creators of digital technology be positioned in some "straightforward" service relation to scholarship? Conversely one may ask: how much scholarly expertise and associated skills should be transferred to the gift bearers of digital technology (i.e computer scientists, software engineers, and scholarly hybrids) to warrant responsible digital scholarship? Is the trading of skills a zero sum game? In all, the question of whether one needs to know how to program to be a digital humanist has been a hotly debated topic for years. My research questions involve how we answer this conundrum after we have taken a much closer look at the relationship between code and hermeneutics.

Narrowing one's research subject is a practical and analytical necessity. But narrowing down is hard. Potentially important aspects may fall beyond the defined scope. In my case it is even hard to determine to what field or domain we are narrowing down. Depending on one's perspective and context digital textual scholarship or digital textual editing may be assigned various positions in the academic landscape. It can be regarded as a subfield of Digital Humanities – a field that itself struggles with the question of whether it is an actual academic discipline, a methodological sub-discipline, or an interdisciplinary middleground between computer science and humanities (Terras, Nyhan, and Vanhoutte 2013). But digital textual scholarship can also plausibly be positioned as a subfield of textual scholarship, or an interdisciplinary supporting discipline for literary criticism and historical studies. I chose to consider the impact of digital technology only to the extent that this technology impacts the methodology and mechanism applied when creating scholarly editions, but even from that perspective my work is highly selective and narrow. I specifically and intentionally do not focus on topics like digitization,[2] digital sustainability, knowledge infrastructures, or the avenue for textual scholarship to take born-digital culture into its domain of enquiry. However, it is unavoidable to touch upon some of these aspects very generally at times.

My primary practical intention is to study and critically reflect on the experiences in a number of digital humanities projects at the Huygens Institute for the History of the Netherlands. I derive my methods of reflective study from Science and Technology Studies (STS), which is the research domain that studies science and technology, and how science, technology, and society shape each other. Through these methodic reflections I try to provide insights into how textual scholarship methodology is affected in both overt and covert ways as a result of the interaction between scholars, digital humanities researchers, and software engineering professionals. STS has not often taken humanities as its object of study. Mostly it has concerned itself with the sociology of the sciences (e.g. Latour 1987) and the relation between the sciences, technology and society (e.g. Bijker et al. 1987). By observing a number of digital humanities projects and their context from the Huygens Institute as "raw data", this study answers to a call of Christine Borgman (2009) that could typically be placed well within the realm of science and technology studies: "Why is no one following digital humanities scholars around to understand their practices, in the way that scientists have been studied for the last several decades? [...] Given how rapidly scholarship in the humanities is evolving, it is fertile ground for behavioral research. The humanities community should invite more social scientists as research partners and should make themselves available as objects of study. In doing so,

[2]Note the difference between "digitalization" and "digitization". The former I use for the process of work and practice becoming more based on digital tools and data, while "digitization" denotes the technical process of creating digital counterparts of physical objects (e.g. creating a digital text file through optical character recognition, or a 3d digital model of a room using laser scanning techniques).

the community can learn more about itself and apply the lessons to the design of tools, services, policies, and infrastructure."

Given the above this interdisciplinary dissertation traverses three disciplines. It finds its subject matter on the intersection of textual scholarship and software engineering, and it studies how these two intersect applying Science and Technology Studies perspectives and method. As a matter of choice and consequence this work also addresses three audiences. Mostly it will speak to textual scholars, both digital and non-digital, who are interested in the confrontation of hermeneutics with the digital scholarly ecosystem. But I also hope to find an audience of technicians working in textual scholarship or the wider digital humanities who may find interest in the knowledge gained from these case studies. Lastly I hope to provide a useful set of studies for the ongoing work in Science and Technology Studies of mapping out the digital humanities and trying to make sense of the socio-technical systems it creates.

Catering to three different audiences is hard. No doubt each will find some matters painfully mundane or explained beyond necessary detail, while other matters may seem difficult and obscure. I can only hope the pain is divided equally and ask my audiences to accept that understanding each other is sometimes hard work. I should also note that I still use the term "computer science" on occasion. Software engineering, natural language processing (NLP), machine learning (ML), and so forth, find their origins in computer science indeed. However, during my investigations I have come to the conclusion that their applicational guise in textual scholarship comes mostly from people trained in computer science who turned to software engineering for a career, or from humanities scholars who have acquired programming or machine learning skills but have not been trained as computer scientists per se. Thus, where one reads "computer science" it is probably best understood as "software engineering" or better yet, as "research software development", which is an emerging field and umbrella term.

At a more concrete level this dissertation tries to answer the question "What is the relationship between software engineering and textual scholarship?" It does however answer this question only in a number of tentative ways, based on selected use cases. These use cases are situated in the context of The Huygens Institute for the History of the Netherlands and are tightly linked to the digital scholarly work that has been going on there for more than a decade. A more theoretical question that it tries to answer in particular is what influence code and code development have on the hermeneutics of textual scholarship. An overarching and, in my opinion more important, meta-issue here is to define (and claim) a certain intellectual space that is located between research software engineering and textual scholarship. Following Galison (2010) the room between software engineering and textual scholarship is often pictured as trading zone, and a gap that needs to be bridged, which marks this place as owned or wanted by nobody in particular. It is mostly pictured as an interface where engineering practice and intellectual humanities work meet. This by itself is a loaded depiction that seems to mostly insist that engineering is material labour and humanities an intellectual one (cf. Burgess and Hamming 2011). The meta-message of this dissertation is that this space should be redefined as an interdisciplinary intellectual academic space in its own right where three distinct types of knowledge intersect: Science and Technology Studies, textual scholarship, and research software engineering (mutatis mutandis computer science). The chapters of my dissertation are examples of one type of intellectual work that may take place in this space of knowledge creation. In particular I make this claim because it has been these three perspectives combined that allowed me to uncover, in the end, a formidable computational challenge for digital textual scholarship.

1.2 Background

1.2.1 Origins and Motivation

Since 2001 my work has always in some way been interdisciplinary. Humanities research intersected with research software development in all the projects I have been involved in. In most cases it was literary research or textual scholarship intersecting with software engineering to apply techniques from data management, information science, natural language processing, and so forth. In the academic humanities contexts in which I worked the application of these new digital methods often caused friction between those whom we might call the technicians and those who were the humanities researchers. To be clear, the technicians were not necessarily computer scientists. They were often software developers or else people like me, who had a formal training as a humanities researcher and next to that training (either formal or through professional experience) as a software engineer. I think two experiences best describe the kind of oppositional perspectives that caused frictions between these professionals. The first was the clearly perceptible hostility we, as developers, experienced from the editors of the renowned Bibliography of Dutch Literature and Linguistics (BNTL)[3]. This was 2004 and offering full-text search for resources via the Web and sorting search results based on computed relevance (i.e. similarity ranking based on some text distance measure) was still a novelty mostly pushed by Google and welcomed with mixed feelings in academic domains. Through tireless explanation the development team tried to “sell” the novel ranking approach to editors – to no avail, however. The other experience was a passionate and eloquent talk given by Annamaria Carusi at the Royal Netherlands Academy of Arts and Sciences on the occasion of a launch of another project I headed. In her talk on digital humanities Annamaria noted the remarkable fact that computer science after six decades of research had only predicate logic to offer: a type of reasoning that the humanities had known for about two and a half thousand years.

Interestingly nobody in these cases was wrong. These are not matters of opinion or fact, but of different attitudes and skills in handling information. My personal experience is that the incongruities and misapprehensions between computer science-inspired work and work rooted in the humanities have led to many frustrating situations in many unproductive research projects. Unwittingly and unintentionally, I may very well have even been a cause of such frictions. Many people involved in these interdisciplinary projects told me in private conversations of an infuriating, off-putting, and seemingly unbridgeable gap between sides. Those stories came from humanities researchers – even those actually eager to engage with new technologies. But they came in equal measure from tried and tested project

[3]http://www.bntl.nl/bntl/

managers, software developers, and department heads. Not even directors were excluded from frustration. The humanities and research software development hybridization looked more like one big unhappy mess and misunderstanding than the makings of a creative and productive interdisciplinary field. This left me with a wish to understand why it is so hard to bring these worlds of different methodologies together fruitfully. As iterated above, it was Anne's remark that set me to thinking to what extent positivistic solution-oriented computational approaches were irreconcilable with the hermeneutic perspective of humanities.

I have come to believe that an important cause for most of the frictions is the way digital methods impact and change interpretation. If it is unclear for software developers what is central to textual scholarship, then it is likely that any digital technology delivered will be somehow mismatched. As long as there is no shared understanding between technologists and scholars of the centrality of interpretation, there will be misinterpretation, miscommunication, misunderstanding, and enduring friction. The BNTL bibliographers preferred a stable representation of interpretation, no ephemeral ranking. The software engineers preferred to delegate some interpretative moves to algorithms. Thus the root of conflict was in a misaligned understanding of who has the authority to interpret, how interpretation should be done, where interpretation ought to happen and how it should be represented. The technologists never realized that their software solution changed the modes of interpretation that were essential to the bibliographers.

Understanding how digital technology changes and affects the methodologies and practices in a certain discipline may best be born from experience and reflection. I started working for the Royal Netherlands Academy of Arts and Sciences in 2000, and I have had the good fortune to be part of a good number of digital humanities projects in various roles since. In 2004 I became affiliated with the Constantijn Huygens Institute, which following a merger renamed itself to the Huygens Institute for the History of the Netherlands. In the period until 2008 the Huygens Institute has claimed for itself a position at the forefront of digital humanities methodology, following its bold ambition to establish itself as a leading research organization in the field of scholarly digital editions and historical research conducted through digital means. Its corporate mission in 2015 stated, among other things, that an aspect making "the Huygens ING so exceptional is the fact that researchers engaged in the humanities collaborate closely with a completely different type of expert, such as specialists in informatics, authorities in digital humanities, and a large team of software developers, all under one roof on a daily basis."[4] As one of the proponents of humanities computing who has been active in the Huygens Institute for over ten years, I find the ambivalence expressed in this mission striking. Researchers in computer science and software developers are a *completely different type of expert*. And digital humanities people may be "authorities" but not all together researchers? However, we should refrain from reading too much subtext into a mere institutional motto.

The Huygens Institute has realized high impact digital humanities projects. But not all high impact projects are equal. The electronic edition of Vincent Van Gogh's correspondence[5] is probably the most widely known production of the institute. In the interest of precision: the scholarly content of the edition of Van Gogh's correspondences was prepared by scholars and art historians from both the Huygens Institute and the Van Gogh Museum. The Huygens Institute also contributed the digital component in the "digital editing" part, which consisted of digital humanities expertise – most notably through my colleague Peter Boot – and the bulk of the immense website development work that went into the project. The Van Gogh project enjoyed the full support of management, as can be expected in the case of a name as well-known as Van Gogh. The story of CollateX is rather different. Now arguably one of, if not the most advanced scholarly text collation engine in the world, the development of the core algorithm and many of its integrations have actually had very little support from the institute. Scholarly input from an international community was guaranteed through a successful grant application that launched the Interedition[6] project. But the specific funding scheme of COST[7], which is aimed at networking researchers and integrating research activities, does not provide for actual research or development work. Over time the support of CollateX has become an open source community en-

[4] https://www.huygens.knaw.nl/organisatie/over-het-instituut/?lang=en (since website changed: https://web.archive.org/web/20150406013931/https://www.huygens.knaw.nl/organisatie/over-het-instituut/?lang=en)

[5] http://www.vangoghletters.org/

[6] http://interedition.eu/

[7] http://www.cost.eu/

deavor[8], but the early stage reality was that the lead developer created most of the core code in spare time during his train commute.

There are counter narratives to the computational success stories too. Humanities researchers at the Huygens Institute, certainly also a number who explicitly seek to further their methodology by computational means, have complained about and bitterly lamented the seeming rigidity and uncooperativeness of the IT department. They seldom seem to get what they actually want. Methodological mismatches are rife. In one egregious case researchers resorted to extracting data from purpose-built databases because the developers did not seem to be able to provide them with the right kind of analysis or visualization. The ever-enduring scarcity of development capacity is obviously not helping – notwithstanding that with a sixteen person software development team the Huygens Institute boasted the largest IT group to my knowledge in any humanities institute in the world. Symptoms like these surfaced even more strongly after a merger in 2011 which doubled the institute's personnel numbers. So much so that management decided that the methodological debate and internal computer science and humanities collaboration should be supported more explicitly through a program of discussion and knowledge networking.

The Huygens Institute and my experiences there should thus provide ample use cases for this study. But I still need to be selective within this context. At times tensions emerge because the interdisciplinary digital humanities research momentum disrupts the structures of an institutional organization managed along the conventional boundaries of academic disciplines. In the Huygens Institute there has always been a clear divide between scientific departments of history or literature studies and a supporting IT unit. Several mergers on, this divide has only deepened. However, organizational dynamics, institutional politics, funding schemes, social setting, personal professional values, and so forth, are not specifically a topic in this study. The main focus is on methodological changes that affect textual scholarship due to the fact that digital tools and data are finding their uses in this field too.

The intersection of textual scholarship and digital technology has grown into a large potential subject area by itself. A concise (and necessarily

[8] Cf. http://collatex.net/ and https://github.com/interedition/collatex

selective) background on this intersection will be useful. Studies usually refer to Roberto Busa's *Index Thomisticus* as the first encounter between textual scholarship and digital technology (e.g. Rockwell 2003). As early as 1949 Busa and IBM collaborated on computerized text analysis for the inference of a concordance of the compiled works of Thomas Aquinas. Since that time the encounters between textual scholarship and digital technology have grown steadily, though for many textual scholars only the arrival of the personal computer and text processing, and later the arrival of the Web, will have resulted in the first tangible interaction with digital technology.

Textual scholarship is no stranger to interactions with technology. The material limits of book technology have been instrumental in shaping the way scholars from ancient times until today have developed conventions that steer the use of the book as a scholarly "machine of knowledge" (McGann 1995). An example of this is the formalized classical *apparatus* that scholars put at the bottom of the pages in scholarly editions of classical works. These works may have been copied many times over and errors and deliberate changes altered the texts through the ages. Such textual differences are minutely recorded by an intricate system of *sigla* and annotations in the apparatus. This is a clever and very specialized system to record as much information as possible about all known (and even postulated) witnesses to a text within the confines of a single book. The digital medium has eradicated these material limits of the paper codex. But that is not to say that text in the digital environment is not material. On the contrary – digital encoding just adds a symbolic layer that maps one representation of text (e.g. human readable text on a display) to another (e.g. binary code in a computer's memory). In the ASCII "paradigm" for instance the character "a" maps to the binary value `01000001` (sometimes prefixed with "`0b`" to prevent misinterpretation as the decimal number one million and one). The zeros and ones of that binary value can be inscribed in solid state or magnetic storage, and such storage in the end is just as materially bound as ink and paper, as Kirschenbaum (2008) has argued. In the case of magnetic storage a zero in working memory will be stored on a hard drive by magnetizing a tiny region of its surface to one of the two naturally possible two magnetic polarities. Thus the essence here is not that the digital environment eradicates materiality, because it does not.

Rather the digital environment has facilitated the manipulation of atomic units of symbolic encoding. All symbolic information (and a great deal of non-symbolic information too) can be encoded and represented through series of combinations of just two states. Any two states, they do not even have to be the famous 0 and 1 dichotomy that is usually associated with digitality. The flipflop – the electronic circuit that represents either of two states in a computer chip – functions because it can charge either of its two sub-circuits, but not both. Whether the one sub-circuit represents a 1 and the other a 0, or respectively + and -, or { and }, or black and white, is purely a matter of choice: a mapping to meaning that a human interpreter decides to settle on. Therefore for anything that we can think up a representation for, we can express that representation in this atomic language of binarity. This is such a powerful principle because, in a computer, both data and process are expressed through the same binary language. That is: both the text we want to represent as well as the process, and symbolic means of representing it are expressed in the same atomic units. The data, for instance, that represents the information of the text on a binary level is turned into readable pixelized characters on a screen by computer code that itself is expressed as series of zeros and ones.

The boundless representational freedom that comes with this may be deeply disruptive for a textual scholarship that, over centuries, has developed its own constraints-based formalism in what could be called a dialogue with the technology of the printed book – a formalism that captures knowledge and expresses or at least symbolizes the status of academic quality of that knowledge. The sheer manageability and production process of books limit their size, and the amount of information that can reasonably be packed into them, while ensuring affordability and practical use. The page-based layout forces static representation, and it invites linear reading and 2D representation. The materiality of the book has contributed to scholarly conventions that signify the boundaries of trustworthy textual criticism. This specific type of materiality is nonexistent in the digital environment. It has to be recreated if one wants to keep to the conventions it realized and expressed. The process of recreation is expressed in code that when run on a computer results in that specific representation. The problem of course is that any other representation is created just as easily with such code. Any conceivable representation of a text that one might be able to think of can in principle be visualized by software code.

Consider a simple example of text coloring. Using specific colors for specific functions in print publication is costly and error-prone – essentially because the process of typesetting and printing is material and labor-intensive. In principle however, one could, for instance, print all keywords in a text in another color than the rest of the text. Of course one could argue about how this influences the readability and even the semantic expression of the text, but that is not the point in question. The point is: once the text is printed that way, it will remain that way, and it will even resist change due to the effort involved. Changing the color and correcting print mistakes requires a reprint, which is about as costly and error prone as the first print run. This is in contrast to digital web publishing, where once the code for representing the digital text is written it requires only a change to the variable that holds the color code for index words from the value for a red to the value for a blue shade. This – assuming that the text has been stored in XML and the lay out "paradigm" is based on an XSLT, HTML, and CSS "stack" – only requires that one, for instance, changes the CSS designation ".keyword: #911" to ".keyword: #119". Do not be worried if you do not know what technical terms like "stack" mean, because for the matter at hand it is the least interesting bit. The salient point is that changing the color of all keywords in a very large text only requires shuffling a few digits in one number representing a color code.

Expressed in terms of epistemological paradigm, textual scholarship is confronted with a situation where its paradigm is not intrinsically co-shaped anymore by the materiality of its preferred carrier for knowledge. The production of its data, i.e. mostly text, has turned virtually overnight from the context of the book to the context of software. Computerization creates an unlimited space and an unbounded freedom of symbolic and visual expression to represent data. With that comes the potential to express text according to multiple familiar or new paradigms. Therefore, representing text as a book has become a choice now and not a given.

To textual scholars this can be a rather uncanny experience. Insofar as textual scholarship is a set of disciplines that is concerned with all textual issues relating to the humanistic record and cultural memory (McGann 2013) it is conservative on principle. Although scholarly editors acknowledge and champion the need for pragmatism in their profession given the wide variety of textual forms they encounter in historical sources, they have also generally found security in how their "discourse" with the material boundaries of the codex has shaped the formalisms with which they treat the re-representation of historical sources that is their academic endeavor (Greetham 1994). A change towards a knowledge space that is unbound by the material limits of print confronts textual scholarship with the challenge to argue editorial principles for (and maybe also from) such a radical free space.

The change to a digital environment is not just radical in the sense of undoing the bounds of the specific constraints of the book. It is also radical in a performative sense. David Berry (2014:72) writes: "[...] what is also radical about the digital is there is no real separation between data and execution". A direct result of Alan Turing's thinking on universal computing (Davis 2012:146–149) is that software can work to change software. If we can write code that executes, we can write code that executes the writing and re-writing of code. In the realm of textual scholarship this manipulability introduces a new and radical capability. The creation of text, by writing for instance, is always about the externalization of a narrative through symbolic encoding. The act of reading is likewise decoding symbols to regenerate an interpreted narrative. These processes of externalization and internalization have always been unique processes with a static result or source: a letter, a book, a note, and so forth. However, this is changed when text becomes part of the digital environment. On a digital medium text is not stored as glyphs but as an intricate series of bits that may be non-linearly fragmented across its storage medium. Digital text in its most basic form is completely intangible and imperceptible to humans. Several layers of code and abstractions are needed to generate a visualization of these bits that conform to the symbolic system that we require for reading. This complex layer of software is a new intermediary to the process of both reading and writing. It is not "just another way of printing". In the first place, the layer of software is invoked *each time* the text is visualized for display, thus the text is generated every single time when its presentation is requested. So the correct metaphor would be that it is "a way of printing each book each time it is read". More importantly, because this layer exists as software, it introduces a facility for intentional covert manipulability of the text that sits between the writer or reader in a way that has no equivalent in the non-digital situation. This fact means that software turns text-as-product quite literally into text-as-process. It is this software layer that – according to instructions and assumptions of the software engineer – decides what is depicted of the text and how.

Thus, an additional performative aspect applies to digital text, or rather the digital context of the text. Normally software – i.e. the developers of software – will want to ensure as much of a one to one mapping as possible from text data to readable glyphs. But, as pointed out before, the manipulability of code and its performative character creates a powerful incentive for a post-human variant of Barthes' readerly and writerly activities. Related control, responsibilities, and authority shift in part from the human reader, writer, or scholarly editor to the engineer and the coded algorithm.

If part of the creative and performative processes of text production and reception are delegated to an intermediate software layer, some control over these activities may thus partly or wholly be taken away from author, reader, editor, and even developer. Most simply, one can imagine filters that will not allow the depiction of certain content, such as words that might be offensive to a certain audience. Such a filtering mechanism may integrate with the rendering process of a certain text, if the code that takes care of the rendering allows for that, for instance by way of an application programming interface (API). The effects may be benign or malicious. The manipulability of the process of generating a readable representation may result in surveillance or censoring. But it may just as well result in a positive and empowering mechanism that facilitates creativity if, for instance, algorithms were trained to adapt the vocabulary, tone, or structure of a text to certain reading and learning practices and skills. In all cases, however, the process of generating the reader text is no longer neutral to the text. A similar argument can be made for authoring text in a digital environment. But in the interest of brevity it may suffice for now to remind ourselves of the meddling in behavior of the grammar and spelling correction functions of many word processing tools. Only if completely shut off do these functionalities not influence writing process.

Very little meticulous research has been done on the changes that intermediate layers of code introduce in the processes of textual scholarship. However, knowledge of the effects is clearly needed to understand how they affect textual criticism and interpretation. Arguably, little study has been made because the skillsets that are needed to navigate the realms of text and code only very partially overlap. Moreover, in day-to-day work scholars and software engineers are more immediately concerned with very different realizations of code. The scholar is concerned almost exclusively with the semantics of the representational graphics on the screen that result from the code, while the engineer is looking at source code that produces that graphical representation. The aim and skillset of scholar and engineer are different. This prevents them from having any particular interest in meddling with the methods and techniques of each other. This so-called "separation of concerns" is actually a primary software engineering principle (Dijkstra 1982, Laplante 2007:85). But who, in that case, is paying any particular attention to how these layers are connected and interacting? Who is studying this important site where the negotiation takes place between software engineer and scholar about what digital textual scholarship is? A critical and problematizing study of the intermediate layer requires deep understanding of both knowledge domains involved, including knowledge of the particular literacies in use in each domain. It seems only reasonable that at least some textual scholars should take an interest in such a widely used technology that potentially interferes with their authority, control, means and methods of production, and so forth. Being one such scholar, I aim to provide examples of such scrutiny.

1.2.2 A Status Quo on Digital Text in Textual Scholarship

To a field whose primary concern is with such things as reliability, sustainability, stability, conservation, curation, and the memory function of text, the radical loss of book-bound materiality and the manipulative-performative nature of software must feel like two very disruptive effects. Yet they are hardly the only effects of digital technology. Research at the intersection of textual scholarship and digital technology has proven highly productive as to academic discourse. An introductory background is not the place to cover exhaustively the profusion of academic argument raised in this interdisciplinary domain, but some impression of the polymorphic landscape of research will be useful. Before doing so, however, we should remind ourselves of the role of post-structuralist thinking. The scholarly arguments emerging in the second half of the twentieth century from the works of Roland Barthes, Jacques Derrida, and Julia Kristeva among others, have been at least as influential as digital technology, and quite possibly more so. More salient maybe with respect to the work here, we should note that these post-structuralist ideas have been recognized and acknowledged by the scholarly community as a methodological force, much more than digital technology has been. The two developments are related, and they shaped many of the developments of digital textual scholarship over the last decades. George Landow (1994:1) has succinctly put this as: "The very idea of hypertextuality seems to have taken form at approximately the same time that poststructuralism developed, but their points of convergence have a closer relation than that of mere contingency, for both grow out of dissatisfaction with the related phenomena of the printed book and hierarchical thought." Post-structuralism opposed the structuralist notion of patterns in history and culture that often involved complete or absolute philosophies, and subordinate categories ("Dark Ages" vs "Renaissance", "Male" vs. "Female", "Developed" vs "Primitive"). Post-structuralists instead stressed diversity and connectedness, highlighted the subservient, exposed biases, attacked authority and hierarchies. Landow puts the hyperlink in a similarly liberating context: a mechanism that emphasizes non linear and non authoritative reading and writing.

Given Landow's remark it is kind of ironic that the digital technology that is now arguably most prolific within digital textual scholarship is XML, which itself is fundamentally hierarchical. The Extensible Markup Language is a markup standard that aims to separate the concerns of visual representation and content description. The appropriateness of hierarchically structured markup for textual description and representation is one the most intensely debated topics among scholars of digital textuality.

Digital markup models arose from a practical need of IBM in the 1960s for information and document retrieval, which required documents to have metatags that could be used for querying. Charles Goldfarb, a lawyer by training, set out to develop Generalized Markup Language (GML). GML was a metalanguage to define actual markup languages and was to become SGML (Standard Generalized Markup Language). The core idea of SGML is to abstract away from specific typography markup codes, e.g. a type setting instruction embedded in the text such as ".bf roman16" to change the used font to 16 point roman face. Instead SGML "thinks" of documents as constructed from distinctive functional parts such as "head" and "paragraph", and it "tags" these accordingly, e.g. "<head>Chapter 1: Fundamentals</head>". A macro language called SCRIPT would then be used to turn the generic markup into specific typesetting codes. The business case for this is that the added indirection allows for more flexible typesetting as well as higher consistency in typesetting. A "head"-tag can be translated to multiple visual representations through associated SCRIPT macros that will ensure that all headings will be typeset equally consistently in the format of the chosen visualization. It appears to have been Stanley Rice (Goldfarb 1996; 1997; Renear 2004) who suggested that markup could be used to denote structural elements of text, turning the tag structure into a description of the structural make up of a text. This was an essential step in moving markup from a purely procedural perspective into an essentially descriptive one. From his memoir it looks like Goldfarb was less motivated by this structural descriptive ability of markup, just duly noting that Rice's contribution was the idea "that smaller documents could be incorporated as elements of larger ones".

Strictly technically speaking, a hierarchical view of text is not a sine qua non for the descriptive abstraction Rice had in mind. But a hierarchy of neatly nested descriptive elements is a must if such a description is to be computationally checked for well-formedness and semantically validated against some predefined set of description rules. Without a strict nesting (i.e. a strict hierarchy) it is impossible to tell which "</russian_doll>" closes which opening "<russian_doll>" in a deceptively simple looking but complex case example like "<russian_doll>russian doll 1 and <russian_doll>russian doll 2</russian_doll></russian_doll>". This example is only non-ambiguously interpretable under the strict condition that tag pairs must be nested inside of each other. The salient point is that at this point technical constraints started shaping what is essentially a scholarly descriptive semantics. The technical constraints became rationalized only post hoc when DeRose and Renear theoretically elaborated the hierarchical approach into the concept of OHCO, text as an Ordered Hierarchy of Content Objects (DeRose et al. 1990). This model, adopted and promoted by the Text Encoding Initiative (TEI) influenced the later development of XML[9]. Emblematic for the techno-textual discourse at that point is the role of Michael Sperberg-McQueen who combines a PhD in comparative literature with extensive knowledge of markup and semantic technologies. Sperberg-McQueen both heads the working group that proposes the XML specification and is co-editor of the TEI guidelines, which find a wide adoption in the textual scholarship community and can be called a de facto standard for text encoding in the field (Sperberg-McQueen 1994; Bray and Sperberg-McQueen 1996).

The hierarchy inherent in the grammar of XML does not force a hierarchical structure onto the resources it describes. Although it may be awkward and unwieldy at times, it is quite possible to describe non-hierarchical constructs with XML. However, the guidelines of the TEI consortium – a basic ingredient of almost every tutorial catering to scholars wanting to create digital editions – are grounded in such an idea of text structure as an hierarchy. As a result most TEI based XML resources are hierarchical representations of text. This "hierarchical assertion" in the de facto use of XML for text markup, as propagated by the guidelines, has been a topic of extensive scholarly debate. In its most pragmatic and tangible guise this debate is topically known as the "problem of overlap". The hierarchical demands of XML make it hard for scholarly editors to markup coinciding textual features (cf. e.g. Schmidt and Colomb 2009). Such problems of overlap occur even in the context of a single hierarchy. For instance, when the document structure requires the

[9] It should be noted that another major drive behind XML was a perceived defect of HyperText Markup Language (HTML), the first practical and very successful markup language for putting hypertext documents to the Web. HTML focuses significantly on description of lay out with such tags as "<B>" (bold) and "<i>" (italics), that is, exactly what Rice was trying to abstract away from in the interest of generic and structural markup. Somewhat ironically the objective of XML, to describe semantics rather than representation, is now also increasingly implemented by other technologies that are part of the HTML5 specification (e.g. data-attributes) or that can be embedded in HTML5, such as Microdata (http://www.w3.org/TR/microdata/) and RDFa. This is not to say that XML is on its way out, but it seems unlikely that it will replace HTML as might once have been hoped or thought.

text of a page to be enclosed by an opening and closing tag this may conflict with the opening and closing tag denoting the logical structure of a strophe that spans the page boundary. Because XML requires elements to be nested, a problem arises: the page does not contain all of the strophe, nor does the strophe contain all of the page, thus document structure and logical structure cannot be expressed in one neatly nested single hierarchy of elements, unless awkward "hacks" are applied (DeRose 2004). Paradoxically, things may get more complex when different conceptual hierarchies are separated out. One could describe the document structure with a single XML hierarchy, and use a different XML hierarchy to describe the logical structure of the text. However, even if each hierarchy is kept apart, when rendering a visualization based on multiple hierarchies the visualization process still has to solve these conflicts that arise when the same part of the text or document gets adorned with element classes deriving from separate hierarchies. Moreover, as overlapping conflicts may also arise in a single hierarchy, this solution cannot be generalized. As a more generic solution, standoff markup is often proposed. Standoff markup is not embedded in the text itself, but resides outside the text in a different file and points into the file of the actual text to indicate the position of each element's start and end tag. To this end some consistent addressing system is needed, usually – but not necessarily – based on the linear position number of characters in the text. Standoff markup adds considerable additional demands and constraints on systems to keep text and markup congruent and synchronized. Out of the box systems do not exist and the technology is very much in a perpetual nascent stage of development. As a consequence this approach has not found mainstream acceptance and application within the textual scholarship community.

The problem of overlap on the implementation level is a symptom of a number of text theoretical debates that pertain to the appropriateness of hierarchies as structural tenets of describing and representing text. Buzzetti and McGann (2006) for instance have remarked on the "radical insufficiency of the OHCO thesis", arguing that XML markup lacks a recognition of structural mobility as an essential property of the textual condition. My understanding is that McGann and Buzzetti reason the inadequacy of the intention of TEI-XML markup along two lines. First of all, markup as applied in the textual scholarship context actually marks up "the pre-existent bibliographical markup and not the 'content' that has already been marked in the bibliographical object." In other words: the bibliographic code of the physical document is in itself a form of markup of the actual "semantic value" that was to be expressed through the document. Therefore, however intricate the XML markup applied to describe the document, it will necessarily be "only" a description of the document-as-markup of the original intended "semantic value". Secondly, they argue that markup itself is transformational. XML markup is only new in its technical format. The function of markup to signal or guide specific interpretation was pre-existing, for instance in the use of single quotes to designate the non-literal or self-referent use of a word. Thus markup is nothing more – and also nothing less therefore – than a form of the reflective function of text. It is of itself part of this reflective function, and as such markup will reflect the editor's interpretation of the source described more than its very content. The argument of McGann and Buzzetti delivers a thorough rationale to the platitude that the meaning and function of XML tags can be debated. Essentially calling on Wittgenstein, they apply a variant of Heisenberg's uncertainty principle to textual scholarship and interpretation: only by adding an XML-tag can an interpretation be made explicit, but the interpreted text is immediately changed by the tag itself, and consequently text as well as tag are open to new interpretations.

Fiormonte and Pusceddu (2006) argue that writing is not simply transcription of the spoken word, but that any resulting text "also offers us a conceptual model". Analogous "encoding provides us with a 'conceptual' model of the original text obtained by means of 'metalanguages' – the markup languages." Thus XML proposes an "interpretative model of the textual world" that regards text as data and as networked data, given its allegiance to hypertext. But XML's conceptual model is based on a conception of text that is firmly pre-digital, that it is simply inscription and transcription. However, text is also a particular kind of experience. It is not just signs on a page linked to signs on other pages. Next to that, text has a temporal dimension. Even if a text does not change in a literal sense, its meaning is affected by distance in time. "These aspects become more important in modern and contemporary textual criticism, where the attention shifts from the product to the process. [...] If we analyse the work as process and not as text and, above all, if we frame it in a context of interaction with the user/consumer (as it is with certain types of online writing), we can say about writing that which is said about other media: 'that which happens in practice cannot be deduced simply from that which happens in texts and in structures'." The XML models in use do not provide for expressing such aspects of process. Neither do we have any idea of the reciprocal effects of these models, as we have not developed an "adequate theoretical frame for the new relationship, which, in the digital dimension, is established between processes and products." Fiormonte and Pusceddu compare this situation to a pre- and post-Heisenbergian universe. XML is conceived from a worldview governed by stability, in which the observer does not modify the object observed. But writing in the digital context, by means of digitalization, is returned to the status of a process and the XML tools we use are inadequate to express a re-entry of text into this more fluid existence.

The above examples serve to show that, very early on in the development of digital technology for representing text, a mutual reciprocal shaping of technology and scholarly discourse starts to take form, and is continued ever since. The development of XML thus is not autonomously deterministic – that is, only determined by itself and its initial state without being influenced by external factors. It is not a process determined only by technology that uni-directionally impacts textual scholarship. Rather, the contribution of textual scholarship to the principles of XML is extensive and significant. And the development of XML – or text being remediated in a digital context in general – causes new scholarly debates. These debates are informed by different perspectives on text, which in turn are informed by different scholarly needs, technical skills, institutional context, and so forth. Not only text is situated, but what one needs from text and technology is also situated.

Central to all debates, however, is the ever present friction between static and fluid views of text in textual scholarship. The digital environment only adds to the conundrums that textual scholarship faces with regard to this friction. Willard McCarty (2005:26), referring to several computer scientists, argues that the fundamental difference between digital computing and other types of technology is that computers work by manipulating representations. This, as pointed out before, is the nature of the Turing machine as a universal computer (Davis 2012). Alan Turing realized that the logic involved in handling data as well as the data itself can be coded using one and the same encoding. Essentially a computer reads the first part of such a code as an instruction to build a logical space into which data is then brought to be manipulated in and by this logical space. The exact same symbolic representation expresses the two essential components of digital computation: computer language and digital information, logic and data, or in yet other words, process and model. This distinction between process and model, between the study of manipulation and of object, is mirrored by a number of current debates in digital textual scholarship that could be categorized under a "text-as-[fill in a particular metaphor] argument" label. It is arguable to what extent these debates have been caused directly by the digital remediation of text, but they have certainly all emerged from textual scholarship undertaken well within or at the boundaries of the digital environment.

One example is the renewed attention for the idea of text-as-object by reinvigorating the emphasis on the material and documentary nature of text, which is a result of – inter alia – the perceived poor ability of digital technology to represent materiality. Scholars like Jerome McGann (1991; 2001) and Donald McKenzie (1999), though the latter is less pre-occupied with digital aspects, have called attention to the relevance of materiality to questions of interpretation and the function of text. McKenzie's argument is essentially that materiality is far from neutral to the meaning and interpretation of a work. A book and a movie can mediate the same story, yet their material makeup induces a specific mode of storytelling and sets conditions and constraints for the interpretation of the narrative. McGann points to a similar essence of materiality within the textual domain itself. He refers to works of William Blake, who put text to the page as and in conjunction with pictorial material, such that representing these works as text-only editions boils down to severe misrepresentation. Another example of the "extreme" materiality of codex technology is the knowledge representation we find at the fingertips of medieval writers and commentators. The formalization of the intricate system of comments, annotations, and Tironian notes in the margins and line spaces of medieval codices was shaped importantly by negotiating the material limits of the codex – quite similar to the ways the classic apparatus was formed. These codices and their makeup were indeed technologies for knowledge management (Teeuwen 2011). The ramifications for textual scholarship of a poorly remediated materiality could be severe. The human mind accepts tangible tools as an extension of the body (Cardinali et al. 2009). In a similar fashion McGann has argued that the codex as a "machine of knowledge" is an extension of the human mind (McGann 2013). The relation between our technologies and us is mutual formative. We shape technology, and technology shapes us (Capurro 2010; Berry 2014). The question of how the interplay of digital technology and textual scholarship shapes the interpretation of text-as-object should thus have obvious importance for textual scholarship.

As shown earlier, scholars such as Buzzetti (2009) and Fiormonte and Pusceddu (2006) have brought the perspective of text-as-process more to the forefront than a perspective of text-as-object. They draw attention to the processes that pertain to text and how the digital context changes, enhances, or adds to such processes. Reading, writing, the editorial workflow, the output of publication or data, the reception, the engagement with text, and so forth – all these processes change when they are executed in a digital environment. For other scholars the notion of text-as-process in combination with digital technology opens up venues to explore the nature of scholarly editing (e.g. Sahle 2013) and that what McGann (1991) called the textual condition, which is the inescapable truth that any act of editing results in a changed, and thus a new text. Johanna Drucker and Jerome McGann, for instance, designed the Ivanhoe Game, which is an application that in a turn-based game style confronts editors with the consequences of their editorial interventions (McGann 2003; Drucker 2003). It visualizes the intricate interaction and influences that editorial decisions have on text and on each other – in turn allowing scholarly editors and textual researchers to investigation and reflect on their methodology.

Another contingent of textual scholarship understands text-as-process to mean the use of digital technology to involve a broader audience in the editorial process. This body of research is self-consciously "decentering" the scholar and editor and embracing ideas of open science and community engagement. Its critical scholarly program, with "critical" understood both in the sense of textual criticism as well as in socio-political terms, is probably most succinctly expressed by Peter Robinson (2004): "All readers may become editors too". Digital technology impacts the scholarly digital edition and scholarly editing in many ways, but a profoundly advocated effect seems to be the movement towards a more open scholarly process, involving for instance communities of interested non-professionals by crowdsourcing elements of the scholarly workflow (Brumfield 2013; Causer and Terras 2014), by adopting open science characteristics (Shaw, Buckland, and Golden 2013), and by theorizing the social edition (Siemens et al. 2012).

The "process" aspect in the text-as-process perspective can also be taken to stress the computational handling of text and the modeling aspect or process that is involved with that. Willard McCarty and Julia Flanders, for instance, treat the development of digital scholarship from a vantage point of the philosophy of science and try to determine how the digital process affects humanities scholarship methodologically. For Willard McCarty (2005) the essence of process is not so much the computational process, but the heuristic of the digital scholar by which that process is changed and adapted to evolve a computational model. That model itself cannot be perfect, since by definition a model never is. But the point of making the model is to reveal new knowledge through observing the differences between model and reality. Julia Flanders (2009) also finds essential value in the friction between model and observed reality: "representational technologies attempt to restate those [humanities] methods in terms which are not identical to, not embedded in the humanities discourse. They effect a distancing, a translation which, like any translation or transmediation, provides a view into (and requires an understanding of) the deep discursive structures of the original expression." With a perspective on process in which new knowledge is uncovered mostly by the inadequacies of the (computational) model McCarty and Flanders align themselves very much with pragmatic phenomenology, the philosophical paradigm which was heavily influenced by the ideas of Martin Heidegger (cf. Coyne 1995).

Stephen Ramsay, however, rejects the idea that the most valuable computationally created knowledge derives from actual computation breaking down. Ramsay rather pictures the algorithmic process as an adequate tool to support hermeneutic investigation and literary criticism. A computer or algorithm can not infer a "Marxist reading", but it can, given a training set of examples, trace such features through a vast corpus unerringly (Ramsay 2011c:34–35). Ramsay thus demarcates the boundary between what could be called the hermeneutic–positivistic divide in digital textual scholarship. The more extreme side of the computational spectrum can be associated with empirical quantification. Here one encounters scholars such as Franco Moretti (2013), whose ideas on distant reading seem to implement to a certain extent the algorithmic criticism envisioned by Ramsay. Even further along that spectrum we find, I think, the works of Burrows (2002), Piper (2015), Kestemont (2012), Underwood (2016), Van Reenen (1996), Andrews (2012), and a large number of other scholars well embedded in the fields of stylometry or stemmatology, traditionally the fields in textual scholarship that have been relying most heavily on computation. Arguably, one could categorize this direction in digital textual scholarship under the label of text-as-data. The associated hermeneutic–positivistic divide in the digital humanities seems not to be a very deliberate division based on fundamental scientific philosophical choices and allegiances. It appears that some researchers – such as Johanna Drucker – are interested in the hermeneutic tenets of humanities and how they should be digitally remediated (cf. e.g. Drucker 2011), while others, on the more computational side of matters, simply are not. But virtually never are hermeneutics explicitly discarded (e.g. Bod 2013a:333–334). It would therefore be too easy to accuse researchers on the far end of the digital humanities spectrum, using statistical methods and computational approaches, of being motivated by positivistic scientism and technological determinism. At worst it is, quoting Johanna Drucker (2010:36), "naive empiricism".

Meanwhile, yet another contingent of textual scholars seems to strongly oppose the text-as-process perspective. Rather in contrast they forcefully reassert the stability of the codex. Elena Pierazzo (2011) and Hans Walter Gabler (2010), for instance, seem to view the digital environment foremost as a space that enriches our abilities to convey as exactly as possible our understanding of the text-as-document. This line of textual criticism is strongly associated with the TEI markup approach and its ideal seems to be to mimic as closely as possible the source that is in front of the scholarly editor: to augment the immutability of the codex as the witness of a text.

1.2.3 The Hermeneutics of Digital Textual Scholarship

The debates on digital text in textual scholarship seem to focus on representation. What is remediated digitally? How is this remediation done? How is the result presented on screen? But the question of what this remediation means for interpretative matters seems not to be raised at all. This is what interests me. First of all, for instance, who is doing the interpreting? Perhaps a silly question at first sight, for everybody interprets. But the "who" is actually substantially changing in the digital context. Projects in digital scholarship tend to be interdisciplinary or multidisciplinary. They reach beyond the confines of the desk of the individual scholar, and beyond his or her skills. Digital textual scholarship turns the work that was once predominantly the work of few into a work of many. However, this trend is not reflected in a broadening of the authorship of what is considered academically viable publication (Nyhan and Duke-Williams 2014) nor are the many included in the process of distributing academic credit (Borgman 2015). Moreover, software is not routinely evaluated, and code work is neither credited nor accounted for. Software engineer, computer scientist, computer linguist, hybrid scholar, web editor, data curator, and so forth, have all become part of the scholarly process, and it strikes me as strange and suspicious that no one is asking what these roles and their actions contribute with regard to the process of interpretation. Nor does anyone ask: how does software code relate to or affect interpretation?

This question of how hermeneutics is impacted by the application of digital technology in the humanities is as urgent as it is important. Digitality – or softwarization as Berry (2014) calls it – permeates all aspects of society. There is almost no workflow that is not executed at least in part in a digital environment. This impacts the humanities in two ways. Firstly, it changes how society, culture, history and their artifacts are created and perceived: in other words it affects the research data of the humanities. A new form of data is created by the digital environment itself, e.g. data streams deriving from Twitter, but also "physical" data is affected when it is digitized. Secondly digital methods affect the means by which the humanities treat and analyze these data.

There is a high degree of covertness in the way softwarization permeates society and thus aspects of human communication, behavior, and culture. It settles as an almost perfectly transparent layer of code and software between physical objects, data, or information and the users of these objects, data, or information. Almost all information that people use in everyday and professional life passes through this invisible layer. To the average user, digital infrastructure is indeed as transparent as the infrastructure of tap water. The diffusion of this layer into society is now so ubiquitous that it is no longer regarded as an alternative context next to society: rather, there is a full commingling of the context of society and the virtual context. Following William Gibson, the author of the cyberpunk novel *Neuromancer*, Steven E. Jones has dubbed this process the "eversion" of cyberspace: a turning inside out, the unfolding and emergence of the digital environment into so many contexts of the real world, for instance through mobile computing, that both become intrinsically intertwined and inseparable (Jones 2014). Jones argues that the forms of mobile and social computing that boomed in the period 2004-2008 have especially contributed in a decisive way to this eversion. Ironically it appears that the more thorough this eversion becomes, the less remarkable or noticeable the abstract layer of digital software is to users.

These processes are not trivial. As Capurro (2010:37) argues: "We are bodies in technologies. This is particularly true in the case of the Internet. We are (not just) our brains and thoughts (our beliefs and desires). If we argue that the ways we perceive reality and the thoughts we develop are shaped hermeneutically by our digital technologies and vice versa, then it can be inferred that digital technologies have to adapt to the ways we perceive and interpret reality, otherwise they will be useless and, in the worst case, dangerous". Approached uncritically or unwittingly this digital layer may yield undesired results. Richard Coyne (1995) contends that code and software are anything but neutral technologies. He observes how software – pushed and developed mainly by a market of business and industry – expresses predominantly a neoliberal ideology. This non-neutrality leads David Berry to propose that a critical theory of "the digital" is needed: "Our societies are increasingly relying on digital technologies of the form that incorporate computational and therefore calculative and computational rationalities which therefore raise important questions for critical theory" (Berry 2014).

The connotation of "neutrality" of code is arguably due to the fact that code is grounded in mathematics, logic, and electronics. Computers and software carry a seemingly self-evident claim of mathematical precision, correctness, and often even infallibility. There is a difference however between correctness and intent. Berry argues that computationalization formulates and re-inscribes in software what we think is important. It may be more precise, however, to state that code re-inscribes what software engineers and computer scientists think their clients understand as important. And in many cases software probably simply re-inscribes what developers think is important. Computationalization reformulates what topics are important, and it affects the nature of truth claims. In a social context permeated by digital processes and objects, it may be that only that which can be formulated in computationally tractable terms can assert importance. Similarly, truth in such a context is potentially reduced to that which can be computationally proven. In any case it is certain that (the constitution of) truth can be manipulated computationally. If the reader needs evidence beyond "post-truth" politics, she could be pointed to the experiments conducted by researchers of Cornell University to observe emotional reactions in users by changing the feed streams of Facebook users (Kramer, Guillory, and Hancock 2014). In a world of "screen essentialism", as Berry calls it, models, code, and data can be manipulated and changed covertly behind a veritably unchanged graphical interface.

As noted earlier, I do not want to investigate these claims on the scale of societies. Rather, I want to observe, uncover, and theorize based on case studies how these aspects and effects of softwarization affect the hermeneutics of textual scholarship. But I do think my study speaks to two broad and important problems that can be identified in this context.

The first is the already mentioned hermeneutic–positivistic divide. It is remarkable to me that virtually all researchers in digital humanities as well as those in digital textual scholarship only study the end result of computational or digital processes hermeneutically. Øyvind Eide (2014) wrote: "For many scholars in the humanities the focus is on what happens between them and analogue signals, no matter if the signals are created based on digital signals." This I think is true. However, he then added "The ones of us more or a little less focused on what happens while the signals are digital are called digital humanists." That I think is disputable: digital humanists are applying digital code as a means to an end and only the result of computation is subject to specifically hermeneutic scrutiny. Humanities researchers, either digital or not, seem to accept the graphical interface as a hard boundary between the realm of quantification and the realm of hermeneutics. The nature of computation and the digital as quantified and reductive is tacitly or even explicitly asserted, and is in itself never questioned. Furthermore these worlds can not be of mixed character, in the words of Stephen Ramsay: "It is not merely the case that literary criticism is concerned with something other than the amassing of verified knowledge. Literary criticism operates within a hermeneutical framework in which the specifically scientific meaning of fact, metric, verification, and evidence simply do not apply" (Ramsay 2011c:21). Ramsay implies that code and quantification have no hermeneutical character when he continues by contending (as cited above) that a computer or algorithm can not infer a "Marxist reading", but that the result of aggregated data and distant reading may serve to trace features and characteristics of such a reading. Seemingly, code equals the quantitative, the reductive, and the non-hermeneutic. I firmly believe however that this is a misconception. Software code, and even quantified models, have a deeply hermeneutical nature, they have intent and a priori interpretative aspects. I hope the use cases in this work invite researchers in the field of humanities and researchers in digital textual scholarship especially to consider this interpretative nature of code. If hermeneutics is a central tenet of humanistic research it must be very precisely investigated how this hermeneutics translates into digitality and into code, and what the relation is between code, quantification, and the hermeneutic.

The second problem is that of the apparent transparency of the layers of digital abstraction between humanities research data and the humanities researcher. As with society at large an abstract layer of digital code and software has emerged between the humanities researcher and the data that humanities study. But as noted above: code is not neutral and it is not without intent. Software code is made by people, and people have motivation and reasoning that determine how software code is written. The authorship of code is not objective and disinterested. Software developers may intend their code to be hermeneutically agnostic, but more likely it is not. And what is worse:

they probably do not think about such problems at all. In this sense the humanities simply have the obligation to expand their horizon of research into the domain of software development and code authorship. Not doing so would mean that the very research community that is specifically equipped with several centuries' worth of methodological and subject knowledge of critical thinking and reflection on society, culture, and history, would turn a blind eye to a cultural and social influence that has no precedent in history. Arguably however, the majority of humanities researchers is currently code illiterate and therefore ill-equipped to face this growing challenge.

1.3 Organization of the book

The chapters of this book treat the effects of digital technology on textual scholarship in decreasing levels of scholarly detail. To counter the risk of "inherent myopia" in local case studies (Wouters 2004) the chapters of this book will relate the individual case studies to the larger story of the relationship between hermeneutics and digital humanities, which is the subject of the background exposé in this first chapter.

Chapter two traces the roots of hermeneutics (the theory of interpretation) and argues that there is no such thing as interpretation-free textual scholarship. Rather to the contrary, interpretation is the essence of textual scholarship. The philological tradition presumes some objectified ahistorical observational approach. However, similar to the "textual condition" (McGann 1991) there is a "hermeneutic condition", and the vision of an all-encompassing archive of philological fact turns out to be illusory. But if all is interpretation, then how does hermeneutic theory pertain to digital textual scholarship? Capurro (2010) argues that hermeneutics has, in the latter half of the twentieth century, pseudo-critically rejected technology altogether. The result of this is that digital networks now shape us, but we do not so much actively shape the network technology. Collaterally, hermeneutic theory is underdeveloped in digital humanities and digital textual scholarship. A facile opposition is often conjectured between reductive quantitative approaches and holistic qualitative interpretation (e.g. Drucker 2010), but a sincere intimate investigation of software and computer code as a hermeneutic means of textual scholarship – a computational hermeneutics – turns out to be lacking.

The next chapter is a case study towards digital textual scholarship in the context of the Huygens Institute, more specifically a case study of developing a GUI (graphical user interface) based editor for digital textual scholarship. This work leads me to analyze a complex of possible causes that inhibit digital textual scholarship from developing a hermeneutics that encompasses a deeper engagement with code as a tool of textual scholarship. Interfaces are supposed to enable and empower us to leverage the innovative potential of computer code, but the case study suggests that graphical interfaces merely inspire remediation of well-known forms of scholarly engagement with texts. Paradoxically graphical interfaces turn out to be as transparent as they are opaque. They translate digital models into more familiar guises that refer to the user's knowledge domain. But this familiar guise also hides the actual digital model and makes it unknowable. This drives what I came to call "paradigmatic regression": the use of new digital technology exclusively to mimic existing epistemological concepts. As a result of such regression, digital textual scholarship accedes to a representational philosophy that understands text solely as a digital metaphor of the printed book. The same mimetic predisposition drives the particular use of markup languages applied by textual scholars. Although theoretically viable as an expression of textual scholarship, this mimetic stance does nothing to narrow the gap between computational methods (e.g. distant reading) and more hermeneutic approaches. For that common models are needed. Such models might be found, for instance, in knowledge graphs.

Chapter four takes the analytic notion of paradigmatic regression and develops it somewhat further, in particular by evaluating the notion against Galison's metaphor of the methodological trading zone. Galison (2010) uses this term to describe sites of knowledge where disciplines intersect. If some veritable methodological exchange is happening, then do we indeed see some terminological or methodological pidgin develop at the intersection of computer science or software engineering and textual scholarship? In the specific context of a Huygens Institute case study little evidence is found. In the broader context of digital textual scholarship the situation on a theoretical level turns out to be more ambivalent, with, for instance, scholarly notions about the fluidity of text being likely candidates for support or modeling through hyper media. However, even if digital scholarly editions are information spaces, they seem not to correspond to some new theoretical pidgin.

Chapter five asks: when does coding become scholarship? In textual scholarship, but even in digital humanities as a field, the status of code and coding as a scholarly research contribution is still unclear. So far, within digital textual scholarship, the focal point of the interpretative process appears to be the result that is derived from applying code and algorithm. However, to what extent should the code and the algorithm itself be regarded as intrinsic part of the scholarly argument, and under which conditions do code and coding constitute scholarship? If a programmer is able to derive better results for a certain scholarly task based on computational logic, then does the intellectual ingenuity behind that logic not qualify as scholarship? To answer these questions a thorough understanding of the nature of code and its performative dimension is needed. It turns out that ignoring the scholarly component of code comes at the peril of accepting the fallacy of its neutral nature. The scholarly evaluation and peer review of code is thus not a "nice-to-have". Code literacy is a requisite in digital textual scholarship, and by extension the humanities.

Chapter 6 approaches similar questions about the status of code in textual scholarship: where in this regard does the academic responsibility and authority lie, and where is academic credit due? However, now these questions are put in an authorial perspective: what is an author, what is editorial authorship, and how may code and the creation of code change or contribute to such authorship? In trying to answer these questions this chapter delves into the history of authorship, authorial intent, and their relation to hermeneutics. This allows me to put my argument about the status of the digital edition in chapter 4 into a broader and more historicized theoretical context. It also reinforces the finding in chapter 2 that any "archival turn" in digital textual scholarship is merely illusory while a scientifically more responsible approach would be to emphasize the process and value of interpretation. Although post-structuralism often receives bad press, not least from influential scholarly editors, I find that post-structuralism still provides an adequate frame to describe the social construction of interpretation that authors, editors, and engineers create through revisionary authorship. The software engineer is a new actor in this social construct and in the scholarly process of interpretation. He or she wields a still poorly understood and under-scrutinized delegated agency through the performative aspect of code, and the accountability and evaluation of this agency turns out to be problematic.

Taking together a number of more practical strands in the overall reasoning, chapter seven investigates an alternative digital model for text. It is also an example of how hybrid work – work at the intersection of textual scholarship and software engineering – shapes digital technology as a textual scholarship tool based on scholarly rather than technological considerations. Many alternatives are possible for the prevailing "mimetic digital book metaphor" that was first explained in chapter 2. When considering such alternatives it is pivotal to recall that the very materiality of the physical codex has to an important extent shaped the constraints of print based textual scholarship. Moreover, scholarly editions can only ignore the materiality of any original at the peril of reducing the "interpretational space" that the rendering of a physical text provides. Commonly used forms of digital text are however extremely reductive in this regard, also those within the domain of markup. Advancing in a more boundless digital environment may be frightening because of the eradication of all constraints. However, this boundlessness should not be taken as a rationale for paradigmatic regression. Quite the opposite: it is a core task for digital textual scholarship to figure out what models provide both useful affordances as well as the ability to define useful constraints. Graph models turn out to provide a good balance between ultimate modeling freedom, the need for constraints, knowledge integration, and exchange.

The knowledge and reasoning in the previous chapters are based on interdisciplinary work at the intersection of software engineering and textual scholarship. Specifically adopting a Science and Technology Studies attitude and methodology has allowed me to put a meta-perspective on the subject matter that revealed a lot of the interdisciplinary dynamic that shapes how digital technology comes to be used (or not) in textual scholarship. This STS approach also brought to my attention the broad spectrum of additional factors that equally influence how this interdisciplinary work develops. However, being regarded as peripheral non-scientific matters, many of these aspects do not get reported in scientific publications. But contending that these insights are purely the result of scholarly argument alone, would be far from truthful. In the form of an autoethnography chapter eight tries to retrace the experience and knowledge that equally underpins the scholarship reported in the other chapters. This autoethnography necessarily waves any pretension to exhaustiveness. At the same time it testifies to the productivity of broad and reflective perspective offered by STS methodology. The resulting combined reflective perspective on textual scholarship and software engineering (again mutatis mutandis computer science) finally allows me to make an argument for what I regard as a new formidable challenge for computational textual scholarship.

Chapters two through to seven are versions of previously published and peer reviewed articles. Minor typos and linguistic errors have been silently corrected. A number of references have been updated when a work that appeared first as a conference paper has meanwhile been published as an article in a proceedings or journal. Some sharp-witted remarks by my supervisors required minor changes or revisions to formulations or content. Whenever possible I have integrated shorter footnotes of the original articles in the body of the text, leading to some minor stylistic changes; long footnotes that digress have been kept. URLs that break the flow of reading have been consistently put into footnotes except where the particular web content is the actual subject matter. Introduction and conclusion are original work. Chapter eight finally, is original unpublished work. Here I attempt to add an STS inspired meta-perspective to the dissertation work as a whole.

Chapter 2

Screwmeneutics and Hermenumericals: The Computationality of Hermeneutics[1]

> "But as one of my colleagues was fond of saying, humanists came into those conversations as relativists and left as positivists out of pragmatic recognition that certain tenets of critical theory could not be sustained in that environment."
>
> (Drucker 2012)

Can the computer be a hermeneutical instrument? This question is trivial, for obviously the computer can be. As long as there is a human interpreter any object can be interpreted and can therefore be an instrument of hermeneutical activity. So the question is not if, but how. How can the computer be applied as a hermeneutical instrument of humanities? That question is less trivial, but passes over a number of important precursory questions. First of all: Must the computer be a hermeneutical instrument to humanities? Which again leads to the question: What is the role of hermeneutics in humanities? If we can – at least tentatively – answer that last question, we may progress to evaluate whether digital humanities can and must have a hermeneutics. And if so, the question becomes, how?

[1] An earlier version of this chapter appeared as Van Zundert, Joris J. 2016. "Screwmeneutics and Hermenumericals: The Computationality of Hermeneutics." In *A New Companion to Digital Humanities*, edited by Susan Scheibman, Ray Siemens, and John Unsworth, 331–347. Malden, Oxford, etc.: John Wiley & Sons, Ltd. http://onlinelibrary.wiley.com/doi/10.1002/9781118680605.ch23/summary.

2.1 On Hermeneutics

Hermeneutics is the theory of text interpretation. The very root (Greek ἑρμηνεύω, hermeneuō) means to interpret or to translate. According to folk etymology its origin derives from Hermes, the Greek god-messenger. It is in the nature of Hermes not just to use language as a means of communication, but also to be a corrupter of words, relishing in the confusing power of his messages. He is a god of transitions and boundaries. An apt eponym for hermeneutics, thus – interpretation is the transition of knowledge that happens on the boundary between text and reader. Hermeneutics is already referred to in various ways by classic philosophers, but it is Philo of Alexandria who pulls together a first systematic theory which is aimed at uncovering the deeper allegorical meaning of sacred scripture (Ramberg and Gjesdal 2013). Methodologically connected to the pivotal issue of interpreting the texts of the Bible, hermeneutics plays a central role throughout the history of philosophy, humanistic theory, philology, and literary criticism. There are numerous key works in the development of hermeneutic thinking, but one that should in any case be mentioned is *De Doctrina Christiana* of St. Augustine of Hippo (c.400 CE). In his work Augustine unfolds a methodology to interpret the scriptures. But more importantly, in his methodology he connects semiotics – the theory of signs and symbols – to language, and he connects the interpretation of language to a deeper existential meaning (cf. Green 2008). In his theory words are signs that impart cognitive concepts to an interpreter. Just as a natural sign such as smoke signals "fire" to the interpreter, so do words convey meaning as "given" signs of language. The problem is however that this meaning may be literal or metaphorical. The sun may stand for light of day or for light of vision. The existential aspect is raised when Augustine argues that it is the will and intention of the reader that allows her to address the deeper allegorical interpretation.

From Augustine we take a huge leap through humanistic history and we pass over Thomas Aquinas, Dante, Petrarch, Luther, Spinoza, and many other philosophers and scholars whose names and works stand witness to the profound influence of Augustine's thinking, and of the central role of hermeneutics in the humanistic disciplines (Barolini 2007; Marchesi 2011; Ramberg and Gjesdal 2013). We turn to the early nineteenth century and Friedrich Schleiermacher's contribution to hermeneutic methodology. Schleiermacher points to an important aspect of interpretation, which is that it is in part emphatic in nature. A reader is able to understand a text not just because of a linguistic code shared with the author, but essentially also by sharing a human nature. Thus, a part of the interpretation and part of the meaning of a text is not based on what is in the text, but on what is external to the text. Following, broadening, and formalizing Schleiermacher's work, Wilhelm Dilthey theorized that works are constructed from the vantage point of a particular worldview held by an author. The interpretation and understanding of a text therefore involves relating the text to the biographic and historical circumstances of its author. For both Dilthey and Schleiermacher a basic assumption is that the meaning of texts is grounded in the intentions and histories of their authors (Mallery, Hurwitz, and Duffy 1986). But more importantly, they believed that these intentions were knowable to later interpreters through reconstruction. Dilthey however recognized that this reconstruction would be tainted by the interpreter's present worldview. Interpretation therefore could in his opinion not be objective in a scientific sense of establishing facts empirically. But he argued that aggregation of multiple interpretations could lead to valid and more generalized interpretations.

Both Schleiermacher and Dilthey point us to the fact that any interpretation necessarily involves information that is not in the data itself. This may be information that is available elsewhere in the form of other explicit data, texts, and so on. But interpretation also involves the unique cognition of the interpreter, which is tacit. Acknowledging the partly tacit nature of interpretation sets hermeneutics apart from other frames of interpretation such as the probabilistic model of information theory inspired by Claude Shannon.

Around the time of Dilthey's life and work hermeneutics was still tightly connected to philology, which at the start of the twentieth century was very much geared towards establishing texts according to what was perceived as the intent and ideal of the original author. With Heidegger, that was about to change. For the philosopher Martin Heidegger, the hermeneutic process is not a philological tool. Instead hermeneutics scales to an ontological level and becomes philosophical in nature. Interpretation and understanding pertain to all of us as the interplay between our self-understanding and our understanding of the world (Ramberg and Gjesdal 2013). Heidegger holds that interpretation and understanding are to a great extent intuitive operations. Our understanding of the world is largely an immediate and unreflective grasp of what we sense, based on a priori knowledge accumulated from experience. Heidegger believed that this understanding is uniquely subjective. We can only "read ourselves" into a text. A few decades later, philosopher Hans-Georg Gadamer would be less pessimistic and would suggest that a human can transcend his own horizon by being exposed to the discourse and linguistics of others. Later again, Jürgen Habermas and Karl-Otto Apel added pragmatics into the equation – that is, a theory of interpretation and understanding must also take into account the intentionality of linguistics.

Hermeneutics, then, turned from a theory of the interpretation of text into an ontological theory of understanding. It can now be understood broadly as the theory of the processes that turn information into knowledge. As such, the role of hermeneutics in humanities cannot be overestimated: humanities practice is primarily hermeneutic, its main theoretical frame is hermeneutics. Consequently, the way that hermeneutics developed over time has significant ramifications for the epistemology of humanities. According to Chambers (2000) humanities "is hermeneutic, intertextual, participatory, value-laden, context dependent, and relatively indeterminate; there are no hierarchical structures of information, no obvious causal explanations and no undisputable truths of any significance to be found."

The highly relativistic nature of post-structuralist hermeneutics problematizes factuality as veritably factitious. This poses problems for those realms of humanities that are concerned with establishing the concrete humanistic record – for instance in the case of philology, ironically a humanistic pursuit most intimately connected to hermeneutics. Jerome McGann rejects the post-structuralist project of, inter alia, Lyotard and Derrida, informed by Heidegger's philosophy, to replace "traditional science with a science of the unknown" (McGann 2013). McGann reasons that philosophy is rather a subroutine of philology concerned with testing, reconstructing, or falsifying its subjects of attention. The primary concern of philology then is with establishing the archive of what is known or has been known: "Philology is the fundamental science of human memory". McGann (2013:345–346) reduces the impact of post-structuralist hermeneutics to an "after the fact" reinterpretation of established sources: "For the philologian, materials are preserved because their simple existence testifies that they once had value, though what that was we can never know completely or even, perhaps, at all. If our current interests supply them with certain kinds of value, these are but Derridean supplements added for ourselves".

Philology cannot however escape problematic hermeneutics by simply stating that its aim is a factual archive. More often than not, for instance when difficult script is encountered, interpreting medieval manuscript is nontrivial. Thus, even if it poses as merely recording the words glyph by glyph, textual editing involves interpretation. Moreover, a philologist editing a historical text cannot escape actualization without betraying the pragmatics of philology that presupposes making the archive intelligible for a current audience too. A gloss is instrumental in this translation, but therefore also not ahistorical. Any "ahistoric" presupposition of philological hermeneutics is negated by historicality: "Not only is the decision for one possible correction rather than another already interpretation, but the question of which possibilities of correction occur to the philologist, and which don't, also depends upon his own historical horizon. [...] The intention toward the historical meaning changes with changes in the conception of history. [...] Once it has become doubtful that one can experience how it really was, then it is no less doubtful that one is in the position to establish how something was meant once" (Szondi and Bahti 1978).

This severely upsets traditional philology, which "believes itself to be independent of its own historical point of view." Fiormonte and Pusceddu (2006) problematized in a similar vain the temporal dimension of text, arguing that genetic editions also cannot escape fundamental subjectivity: "one might say that up to now we have analyzed the literary text according to the laws of the pre-Heisenbergian universe, i.e. inside a stable system in which the observer does not modify the object observed." Mutatis mutandis this "hermeneutic condition" can be generalized to many subfields of humanities. The study of history, for example, being dependent on a humanistic record as well, is affected similarly.

2.2 The Hermeneutics of Digital Humanities

Does digital humanities have a hermeneutics like humanities does? Given that digital humanities is humanities too, the answer must be yes. However, there seems to be no focused program to uncover the hermeneutics of digital humanities. I want to investigate whether a call for attention to this hermeneutics, if not a specific program, is a necessity for digital humanities. Rafael Capurro (2010) seems to have come closest to calling for a programmatic approach to digital hermeneutics. Capurro states that the Internet challenges hermeneutics because of its social relevance for the creation, communication, and interpretation of knowledge. That is, the Internet makes the creation and sharing of knowledge a more open and social activity. A problem in addressing this challenge is that the last part of the twentieth century saw a pseudo-critical rejection of hermeneutics with regard to technology in general and to digital technology in particular. But it is exactly digital technology, and more particularly the Internet, that has ontological implications or implications for how we are and behave as humans: the Internet shapes important parts of human expression and experience, and conversely humans shape the Internet as a technology by expressing themselves through it. According to Capurro, a resulting problem is that humans only very partially control the network that they shape but that is importantly shaping them. A counterargument could be that individual humans also only very partially control their physical environment, and that moreover the power of control is unevenly distributed in the virtual as well as in the physical environment. However, Capurro's more important point is that the network is shaping us in more fundamental ways than we may realize. Our lives are increasingly expressed through digital technologies that function as extensions of our minds and bodies: we are different on Facebook, and Facebook makes us different in real life too. This raises questions of a particularly humanistic nature, and Capurro concludes that current hermeneutics fails to address these questions that "go far beyond the horizon of classic hermeneutics as a theory of text interpretation as well as beyond classic philosophic hermeneutics."

If current hermeneutics is unable to address such questions, this may explain the relative lack of theory on hermeneutics we find in digital humanities.

The dialog surrounding hermeneutics seems not to have developed fully yet in digital humanities – references to hermeneutics are scant and often at a concrete level of the practice of text interpretation, such as when Katherine Hayles (2012) uses the phrase "hermeneutic close reading." Yet from several paragraphs and sections in the literature the emergence of a debate seems traceable. Like Capurro, Fred Gibbs and Trevor Owens (2012) have made programmatic claims for a hermeneutics of history writing. Their argument concentrates on data. Data has always been used by historians, but the vast quantities of it that become available should mean "that 'using' signifies a much broader range of activities." Gibbs and Owens argue that using data is not the same as fully conforming to the epistemic burden of the statistician. A playful iterative approach to quantitative tools, explorative and deliberately without the complete formal mathematical rigor, can serve to use large amounts of data to discover and frame research questions. Data does not always have to be used as evidence; in a variety of forms it can provoke new questions and explorations. Data analytics need not be by definition mathematical. "Historians must treat data as text, which needs to be approached from multiple points of view and as openly as possible" (Gibbs and Owens 2012).

Like many contributions in the theory on digital humanities, the article by Gibbs and Owens refers to the opposition between quantification and narrative as methodological means. They write about the "epistemological jitter" and "hostility to data" on the side of historians. Their solution to this conundrum is that data can be read as text. This is true, but it is also an unsatisfactory and incomplete solution to the problem. The presupposition of data-as-text reduces the hermeneutical act to a post-processing of what remains of data after the processes of curation, analysis, and visualization. However, those processes of curation, analysis, and visualization have a hermeneutics of their own. The dialog on the hermeneutics of digital humanities cannot therefore simply posit a dichotomy between the quantitative and qualitative, and relegate hermeneutics to a qualitative aspect of interpretation of given data as if this data would not be value-laden and interpreted already. It is along these lines too that Federica Frabetti – like Capurro – has argued that new technologies affect cultural understanding. She proposes a re-conceptualization of digital humanities that indeed transcends an assumed dichotomy between the technical and the cultural aspects. Such "must be pursued through a close, even intimate, engagement with digitality and with software itself" (Frabetti 2012). Thus part of the hermeneutics of digital humanities relates to the hermeneutics of code, computation, and quantification.

A close or even intimate engagement between digitality and hermeneutics has however not been a main concern of digital humanities. Rather, the opposite has been stressed. In a 1995 issue of Literary and Linguistic Computing Lisa Lena Opas-Hänninen writes: "Only where indexing and sampling are concerned does the computer offer useful help in computer-assisted literary studies. So the impact of computer-assisted techniques sets in before the interpretation and evaluation of the text begins." Looking back, Opas-Hänninen's introduction reads like a very careful attempt to avoid stating that computational analysis in the realm of literary studies can go beyond anything but a pre-hermeneutical support tool. Jan Christoph Meister in the same volume – carefully? – formulates that "an intelligent and well-balanced application of literary computing tools allows us to reconcile the two paradigms by measuring and mapping difference in literary structures, and then forwarding them to the ultimate hermeneutic machine, the human mind" (Meister 1995). Both Opas-Hänninen and Meister at the time argued that the hermeneutical potential of digital technology is limited by the fact that "only questions that can be formalized are open to electronic analysis in literary studies and this is why computer-assisted techniques can cover only part of the work of the literary critic in certain, clearly defined areas of application" (Opas-Hänninen 1995). Meister draws a very strong opposition between the numerical and semantic paradigms: the first is connected to computing, the second to hermeneutics. His argument is basically that semantics do not apply in the computational paradigm. Algorithms can manipulate or process objects, but only insofar as they can be formalized and quantified. Computational operations are strictly and unambiguously transformative: "results are effectively nothing but a more or less sophisticated re-formulation [...] of the original data input." But these transparent repetitions and permutations of data are redundant in the semantic paradigm. When it comes to hermeneutics, "only those results that are different, that happen to question the validity or confinements of the procedures which produced them, will ultimately be found to be relevant and noteworthy." In retrospect it is intriguing that Meister did not consider at the time a distinctly hermeneutic consequence of this argument. Algorithmic transformations can in fact lead to identification of results that are different, and thus not "hermeneutically" neutral. Firstly, even a rudimentary indexing algorithm can transform the full text of a book into a list of terms used more than average per chapter, and can subsequently single out the chapter that shows the least terminological overlap. Is this not a hermeneutics expressed through the algorithm? Secondly, we can consider the breakdown of software. As long as the algorithm only transforms data, it may not be a hermeneutical thing. But it may become so when it falters over some input and breaks down or spews inconsistent and unexpected results. This is akin to what we find in Heidegger's work, which holds that only a breakdown in practice leads to theoretical knowledge (Froesse 2006). As long as a hammer is a hammer, it is a hammer; only when it is broken do we consider its function and how it works.

Twenty years on, the consideration of hermeneutics in the digital humanities does not seem to have moved beyond a basic opposition between patterns and narrative, quantification versus interpretation, that can already be discerned in Meister's strong binary opposition between the numerical and the semantic. This opposition often surfaces as an apparent ideological or political opposition between humanities and digital humanities. Stanley Fish for example has qualified digital humanities as just another fad answering to a crisis of legitimization of the humanities (Fish 2010; 2011). Others point to the ideologies and institutional motivations of innovation, which certainly are not neutral (e.g. Piersma and Ribbens 2013). These crises or ideologies, even if they exist, do not relieve us from critically evaluating the ramifications of emerging digital technologies for hermeneutics. In the first place, these technologies are increasingly used to create the humanistic artifacts that are the objects of study in the humanities. In the second place, we are applying these technologies for the capture and analysis of research data. Both of these processes, motivated by digital technologies, affect our modes of interpretation in nontrivial ways. Piersma and Ribbens argue that evaluation of these digital technologies is "even more urgent in view of the frequently implicit claims [...] that technological progress also implies a new historical-scientific paradigm" – a paradigm based on quantified approaches, on computational analysis of big data, and subsequent serendipitous finds in such big data.

From the perspective of hermeneutics, however, the literature in digital humanities does not seem to justify presupposing an implicit turn to a scientific paradigm. Geoffrey Rockwell (2003), writing on the hermeneutics of text analysis, refers to the French eighteenth-century philosopher Étienne Bonnot de Condillac: analysis merely consists of composing and decomposing our ideas to create new combinations and to discover, by this means, their mutual relations and the new ideas they can produce. Rockwell argues that there is no a priori privilege of any procedure for deconstruction and reconfiguration. But a potential a priori for coherence and homogeneity in computational data analysis may have been inadvertently introduced at the very onset of automated text analysis, which is tied to the computationally constructed concordance by Roberto Busa that was commenced in the late 1940s. Concordancing aims to discover patterns of coherence in a text or corpus – in a hermeneutically naive way because it assumes that a word will have the same meaning and weight wherever it occurs. Yet even the algorithmic creation of concordances shows how deconstruction of a text and subsequent reconfiguration leads to a new text, namely the very concordance. But that is just one method of reconfiguration. To escape naive biases we should shed habitual practices and any axiomatic primacy of unity and coherence. To this end Rockwell – following Gadamer and Huizinga – suggests a hermeneutics of disciplined play that privileges experimentation and modeling, rather than a narrow quantified empirics.

Stephen Ramsay (2011c), even more than Rockwell, emphatically denies a scientific paradigm for hermeneutics: "For decades the dominant assumption within humanities computing [...] has been that if the computer is to be useful to the humanist, its efficacy must necessarily lie in the aptness of the scientific metaphor for humanistic study." Ramsay takes the contrary view, and proposes that the scientific method and metaphor are, for the most part, incompatible with the terms of humanistic endeavor and only lead to a distorted epistemology called "scientism." Ramsay follows Gadamer by stating that the hermeneutic phenomenon is basically not a problem of method at all. Hermeneutics is simply not concerned with amassing verified knowledge of the sort that would satisfy the methodological idea of science. Rather, literary criticism operates within a hermeneutical framework in which the specifically scientific meaning of fact, metric, verification, and evidence do not apply. Yet humanities too is concerned with knowledge and with truth, just of a different kind than that of science. Ramsay has also argued that the availability of vastly more digital data essentially does not change the hermeneutic assumptions of humanities. The fact is that there has always been too much information available to synthesize individually in full; the digital age just makes this condition more apparent. But now as ever hermeneutics involves finding a purposely selective and subjective path through too much information. This is the basic assumption underlying what Ramsay (2010) calls the hermeneutics of screwing around. For Ramsay the "screwmeneutical imperative" is nothing more or less than the realization of Roland Barthes' concept of "writerly text", which is the text a reader constructs by reducing all possible meanings of a text to one that is his own interpretation of it.

In the realm of markup, in the digital humanities predominantly represented by the Text Encoding Initiative (TEI), possible scientism seems not to be a very relevant issue either. This may be partly due to the descriptive rather than analytic nature of markup. The hermeneutic dialog within this domain concerns itself more with the issue of multi-perspectivity. Like Ramsay, Lou Burnard (1998) points to post-structuralist ideas: "Texts, and other artifacts alike, are invested with meaning by our use of them, and it is therefore interpretation alone which confers value on them. Small wonder that Derrida, citing Montaigne, takes it as self-evident that 'We need to interpret interpretations more than to interpret things' " Authorial intention, reconstruction, and original reading are concepts that have become unfashionable, Burnard admits, but he follows Dilthey by saying that there "is ample evidence that not all interpretations are equally useful or have equal explanatory force." He suggests that canonicity is in this sense a hermeneutics of aggregation. Burnard also embraces the post-structuralist idea of intertextuality: the reading and the meaning of a text is in part constructed by the references made to other texts. The rationale for markup then is that it provides a single formalized semiotic system that is able to function as an interlingua for the sharing of the multitude of individual interpretations that through aggregation can lead to a critical consensus.

The claim that a single all-encompassing semiotic system is possible, and that technologies such as SGML/XML and DTDs could be an implementation of it, has since been severely contested. Many theorists and practitioners (e.g. Buzzetti and McGann 2006; Fiormonte and Pusceddu 2006; Schmidt and Colomb 2009) find that the single-hierarchy approach to text structuring that the TEI enforces does not fit well with a multitude of possible structural and semantic interpretations. In itself this dialog testifies to the fact that the approach to text encoding within the textual scholarship and digital humanities communities is primarily hermeneutically oriented.

Thus a computational turn does not automatically imply a turn to empiricism and scientism, or a disregard for hermeneutic tradition. Stylometry and the "school" of distant reading (Moretti 2007; Jockers 2013) may lean in their approaches more towards an empiricist or scientistic attitude. This is mainly to be attributed to the intensive use of quantification and – more importantly – statistics in those avenues of research. The work of researchers such as David Hoover (2013), Ted Underwood (2019), Allen Riddell and Karina van Dalen-Oskam (2018), Matthew Jockers, and Franco Moretti is methodologically strongly based in statistics, corpus linguistics, and natural language processing. Those methodologies are numerically inclined indeed, but this does not preclude hermeneutics – numbers of course allow interpretation too. Quantification does however introduce the problem of reduction. Current statistic approaches to stylometry, for example, are based predominantly on word frequencies and co-occurrence analyses of the surface structures of text. But aggregating words based solely on their form usually blinds these methods to more subtle semantic relations such as homonymy, metaphors, anaphors, and so on, that are also hermeneutically important. This does not however discredit numerical approaches as hermeneutical instruments. In fact they may contribute very strong hermeneutical support.

For instance, Mike Kestemont (2012) has shown, using statistical means such as principal component analysis, that the medieval Dutch Arthurian novel *Moriaen* stylistically stands out from the medieval compilation of Arthurian texts that it is a member of. The text forms a much closer stylistic unit with two other texts, one of which is not even an Arthurian novel but a story in the realm of the so-called matière de France pertaining to the culture, court, and principal personae during the reign of Charlemagne. Based on all we know about medieval Dutch genre and literary history this claim would be outrageous, were it not for a 1970s posthumously published work by a Dutch philologist that had already alluded to these possibilities. Most interestingly, that philologist and poet, Klaas Heeroma, based his conjecture on a fundamentally hermeneutic principle: he claimed he "heard" the kinship between the novels. Somewhat ironically, what is now often perceived of as one of the least hermeneutical instruments – number-crunching-based principle component analysis – indicates that Heeroma's hermeneutical "sixth sense" was right.

The example above draws our attention to another problem inherent in current quantified approaches in digital humanities. As Gibbs and Owens (2012) also point out, neighbor joining, maximum-parsimony phylogenetic trees, z-scores and such probabilistic methods that are used in stemma reconstruction, authorship attribution, and various other computational approaches seem foremost to be used as instruments of reaffirmation. They verify authorship, and they confirm canonicity and genre. They do not answer new questions, but rather solidify existing answers. This may very well be a simple sign of a field in development, of relatively immature application. However, if this confirmation bias were a genuine trait of a specific angle on quantified approaches by digital humanities, then again this would set it apart from the scientific paradigm of falsification rather than import it wholesale into the humanities. So far, quantified approaches in the digital humanities also show a relative lack of explanatory power. Stylometry, for example, can tell us – or rather indicate to us – that there are two authors of a certain text (Van Dalen-Oskam and Van Zundert 2007). But it tells us unsatisfactorily little about how and why the individual styles differ. Engaging and uncovering the "black box" effect of such methods could in due course turn the practice of stylometry into the pursuit of a literary hermeneutics – like conventional hermeneutics but with different means.

2.3 The Computationality of Hermeneutics

Quantified approaches and distant reading currently have good press. But we should be careful not to identify digital humanities solely with these approaches. The field is decidedly broader (cf. for instance Alvarado 2012). There is a tendency in debates to reduce the potential of computation to a methodology of quantification. The nature of digital humanities is hybrid, however, and there is not an a priori discontinuity with the hermeneutic traditions. We still maintain that knowledge has an interpretative character – that the state of an object is determined by its context and is dependent on the observer's interpretation. Computer-mediated text turned text into something computationally tractable. Starting with the work of Father Busa, this made the application of quantified approaches to text feasible and practical. Computational tractability, however, does not dictate quantification and a probabilistic approach. These approaches have been inspired by their success in computational linguistics, a field informed substantially by a positivist and structuralist tradition. This tradition holds that knowledge has a causal deterministic character so that the state of any given object is necessarily determined by its prior states. Probabilistics and, for instance, the Markov models that underpin many natural language processing algorithms derive ultimately from such a positivist deterministic philosophy (cf. Vandoulakis 2011). Johanna Drucker unequivocally denied the applicability of deterministic computational methods to problems of humanistic nature: "Positivistic, strictly quantitative, mechanistic, reductive and literal, these visualization and processing techniques preclude humanistic methods from their operations because of the very assumptions on which they are designed: that objects of knowledge can be understood as self-identical, self-evident, ahistorical, and autonomous" (Drucker 2012).

Drucker's quote summarizes quite succinctly the problems inherent in probabilistic approach that can only lead to "naive empiricism" (Drucker 2010). Grounding the bulk of digital humanities methodology in quantification and deterministic reasoning may have far-reaching disruptive implications. Katherine Hayles pointed out that digital humanities as a field may converge towards traditional humanities or diverge from it as its own field, depending on how digital humanities articulates itself with respect to conventional humanities. "The kinds of articulation that emerge have strong implications for the future: will the digital humanities become a separate field whose interests are increasingly remote from the traditional humanities, or will it on the contrary become so deeply entwined with questions of hermeneutic interpretation that no self-respecting traditional scholar could remain ignorant of its results?" (Hayles 2012) Thus Hayles ties a successful interaction of digital humanities with the traditional humanities to the question of how well digital humanities will be able to cater to hermeneutics. The extent to which the hermeneutic approach is fundamental to the humanities is, however, not always well understood. In his recent history of the humanities, Rens Bod dedicates a mere two pages to the concept and history of hermeneutics, in a section titled "Hermeneutics and the anticipatory 'method' " (Bod 2013a:333–334). He disposes of the "method" as being based on guesswork and premonitions. This dismissal might be cast aside as anecdotal were it not for Bod's position as professor of computational and digital humanities, investigating the humanities from both a computational and a historical perspective. Within the dichotomy between patterns and narrative, Bod has decidedly opted for patterns as a primary principle of investigation. Leaning strongly towards a deterministic paradigm, he concludes that inferences can only be valid based on patterns to be discovered in the researched data. Another example of a dialog between the realms of computation and humanities reveals an interesting "computational" perspective on the fundamental importance of the concept of context to hermeneutics: "We do not exclude the possibility that there may be other relationships that can constitute a valid narrative. [...] However, such examples are context-dependent, and not easily generalizable, we therefore [...] limit our focus to the prototypical narrative structures described" (Akker et al. 2011). This quote derives from a project of which the particular aim was to find a suitable formalization for (historical) events and to build narratives – i.e. historical accounts – from these. The statement reveals the clear tension between hermeneutic context-dependency and the thrust towards the generalization needed for computational tractability. The generalization requires events to be formalized or modeled so they can be computationally traced and quantified. Researchers try to escape the problematic hermeneutics by reducing the number and type of relations that events can maintain. But the problem stubbornly persists, because formalizations and patterns are not hermeneutics-free. Just as philological practice cannot escape a certain hermeneutics, neither modeling nor quantification can escape the hermeneutics involved in choosing the basic assumptions onto which the formalizations are founded. Pasanek and Sculley, in their article on "Mining millions of metaphors" (2008), point out that in this respect there is no such a thing as a free lunch: "It is important to avoid the illusion that automated analysis is somehow more objective or less biased than traditional methods. There is no new infallible science of literature forthcoming. As the "No Free Lunch" Theorem states, every machine learning method requires the acceptance of base level assumptions, such as the appropriate choice of distance metric or the shape of the probability distribution underlying the data. These assumptions must, at some level, be taken on faith, and influence the results of automated analysis, just as cultural and theoretical biases influence traditional analysis."

As with quantified approaches, there is a hermeneutics to any formalization. Textual scholars from Bernard Cerquiglini (1999) to Peter Shillingsburg (2013) hold that an edition of a text is not that text itself, but an intellectual argument about it. A digital edition is an interpretation, and in exactly the same sense formalizations and models are interpretations. A simple example for this is a database field, which is nothing more or less than a category label. Category labels, databases, and data models: all are models, necessarily narrow representations of aspects of reality. Confronting any database with reality, one will encounter observations that will not fit to any of the defined database fields. Therefore most data models exclude certain properties of data, which poses problems in a field such as humanities that works primarily with highly complex, heterogeneous, and non-concrete data. To fit the observations to the chosen categories or properties of the model is to fit a subjectively observed reality to the interpretation expressed by the model. The effect is that the chosen formalization imposes a particular interpretation on a set of data that does not really fit, reducing to a certain extent the richness and complexity of the body of information. The quantitative model or data model is an impressionistic primer onto which more interpretation is painted. Thus statistics and models inform interpretative narrative on the basis of formalized reductive interpretations.

If formalizations, models, and quantifications have hermeneutics too, we can concur with Katherine Hayles (2012) when she states that the tension between algorithmic analysis and hermeneutic close reading should not be overstated. Hayles argues that often there is not an opposition but a synergetic interaction between algorithmic analysis and close interpretative reading. She points to the example of what Matthew Kirschenbaum has called "rapid shuttling," which involves a repetitive switching between the modes of close reading and of interpretation of big data analysis results, comparing the interpretations those different modes yield. Ramsay (2011c), when talking about "algorithmic criticism," also points to this recursive interaction between corpus analytics and close reading that can inform humanistic inquiry of texts. What these views share is that the act of interpretation is postponed to a post-algorithmic phase. Only when the computation has been done and the algorithms and number crunching produce visualizations does the interpretative act come to the fore. This type of digital humanities hermeneutics therefore faces outward and away from the computational model, the math, and the code. It interprets only the results of the algorithmic or quantitative phase. But if it is true that algorithms and models have hermeneutics too, then should these not somehow be taken into account in establishing the validity of interpretations done in algorithmic analysis?

David Berry (2012), like Katherine Hayles, does not "want to overplay the distinction between pattern and narrative as differing modes of analysis. Indeed, patterns implicitly require narrative in order to be understood, and it can be argued that code itself consists of a narrative form that allows databases, collections and archives to function at all." Instead of dismissing code and algorithm as hermeneutic domains, Berry is arguing for a more intertwined articulation of humanities and computer science in this respect. He proposes that digital humanities in part should also concentrate on the underlying computationality of the forms held within a computational medium. To "understand the contemporary born-digital culture and the everyday practices that populate it [...] we need a corresponding focus on the computer code that is entangled with all aspects of our lives." According to Berry there is an "undeniable" cultural dimension to computation as well, which points to the importance of engaging with and understanding code: "Understanding digital humanities is in some sense then understanding code." Berry argues that computational techniques are not merely an instrument wielded by traditional methods. Rather, they have profound effects on all aspects of the disciplines because the computational logic is entangled with the digital representations of physical objects, texts and "born-digital" artifacts. But the way in which the digital archive is deeply computational and the ramifications of that computationality are currently not well understood, and cannot be understood without a deep dialog between humanities and computer science. Federica Frabetti (2012), reasoning along similar lines, concludes that such "an understanding must be pursued through a close, even intimate, engagement with digitality and with software itself" – which is not without problems, as digital humanities and computer science have no readily available mutually informed way of examining software, and because it is: "especially difficult for those not active in the field of the digital humanities to see how the creation of digital surrogates of analog materials, the development of tools to support visualization and analysis, and the contribution of high-end computing skills [...] constitute research" (Schreibman, Mandell, and Olsen 2011).

In the domain of textual scholarship Elena Pierazzo has drawn attention to a similar need to understand coding intimately. Like others, she holds that editing a text is "interpretative and irreversible." She follows Claus Huitfeldt and Michael Sperberg-McQueen in stating that a transcription of a text consists of "a systematic program of selective alteration." Thus it is very unlikely that two scholars, even given the same transcriptional criteria, will produce the same transcription of the same exemplar (Pierazzo 2011). As scholarly editing moves into a digital environment, computational approaches and programming acquire substantial roles and responsibilities in the creation of digital scholarly editions. Pierazzo therefore argues that this role of programming should not be underestimated, and, more importantly, "neither [should] its implicit scholarly content." Coming from a different angle but reaching a similar conclusion, Alan Galey and Stan Ruecker (2010) call attention to the design of artifacts as a critical and hermeneutical act. They argue that digital humanities must not lose sight of design as an act that shapes the meanings of artifacts, and that is no less vital to the interpretative potential of digital artifacts. Galey and Ruecker draw an analogy between software design and the textual and material design involved in book production: "By understanding how fields like book history take the design decisions embedded in physical artifacts as interpretive objects, we can begin to see digital humanists' creation of new digital artifacts as interpretive acts." Digital humanities as yet lacks a deep understanding of digital text production and software design, whereas we have a well-defined understanding of the roles of non-authorial agents in print and manuscript book production, such as scribes, binders, typographers, compositors, correctors, and illustrators (Galey and Ruecker 2010).

The choices and methods involved in software design do shape the hermeneutics of digital humanities. Modeling encompasses the worldview of the model designer, her context, and her subjective decisions. Data models are anything but neutral – on the contrary, they are a purposefully specific selection of semantic categories and properties. Programming languages have paradigms that affect hermeneutics. Moreover, the reciprocal shaping of the hermeneutics of digital humanities by the methods of computer science extends beyond software design. The choices made in the analytical conception of any given digital humanities project affect its hermeneutic makeup. The choices of what properties to quantify, what probability distribution functions are chosen, which statistical tests are used, are in essence hermeneutically informed. Arguably these choices are currently in large part left implicitly to the experts and professionals of software design and computer science. Computer science as a field, however, is grounded not in a problematizing paradigm but in a problem-solving one. Computer scientists and software engineers have a strong generalizing proclivity. Their reasoning tends toward the inductive: solve a specific problem in a specific context and then scale the solution to general applicability. This propensity invites positivistic reasoning and reductive determinism that favors patterns and relegates the exception to the status of "corner case." These characteristics fit poorly with a humanities that is accustomed to reasoning from heterogeneous information, that favors multi-perspectivity, and that problematizes as a means to create knowledge, perspectives, and understanding. The eventual articulation of digital humanities with respect to conventional humanities – and the implications for the future that Katherine Hayles described – will depend to a great extent on how well the intimate dialog between humanities and computer science as discussed by Frabetti and Berry is established. As Galey and Ruecker showed, little attention is currently paid to the hermeneutical implications of the software design aspect. Similarly little attention is given to the hermeneutical implications of data modeling and of analytical models applied in computer science and other fields that inform the digital humanities, such as mathematics and artificial intelligence. Thus at a very fundamental level and in a substantial part of its research chain the hermeneutics of digital humanities is driven by software designers and computer scientists. This means that in practice the hermeneutic choices of digital humanities are made substantially by software designers and computer scientists. Failure to reflect critically on these choices may all too easily lead to a naive scientism permeating the digital humanities, born from the generalizing and problem-solving nature of computer science and software engineering.

Stephen Ramsay (2011c) argues that it is possible to make algorithmic procedures conform to the hermeneutical methodology of humanistic critical inquiry without transforming the nature of computation. Be that as it may, this conformity will not come about without a fundamental dialog between humanities and computer science – a dialog that is not part of Ramsay's hermeneutics for digital humanities, focused as this is on post-algorithmic acts of interpretation. However, a substantial part of the specific nature of digital humanities hermeneutics arises exactly from the nature of computation. This nature need not be reductive, deterministic, absolute, and quantified, as is so often implied. Rather, we have here a rationale for exploring "hermenumericals," a hermeneutics of computation that could complement Ramsay's post-algorithmic "screwmeneutics." Computation need not be a domain of absolute numbers and binary logic. In the field of artificial intelligence, non-binary reasoning and expression of uncertainty has progressed considerably (cf. Russell and Norvig 2009). There are subtler computing logics than the first-order logic that currently makes up the bulk of commonly used computer languages (cf. Forbus 2008; Pratt 1976). Some are concerned, for example, with modeling intuitive notions of truth and validity. Their nature may be a much closer fit for the hermeneutics of humanities. Exploration of the hermeneutic potential of computation is a challenge that digital humanities could pose, to itself and to computer science on behalf of the humanities. This need not imply transforming the nature of computation, but it must involve remediating the nature of hermeneutics through computational logic and design, informed strongly by a dialog with humanities. It is apparent that scientific methods deriving from the humanities would be more appropriate than scientism for artificial intelligence and computer science when interacting with the humanities (cf. Mallery, Hurwitz, and Duffy 1986).

2.4 Conclusion

Unquestionably there is a role for hermeneutics in digital humanities. Thus the question becomes: What does such a hermeneutics look like? Capurro has shown how profound the ontological implications of digitality are for cultural dynamics and for the creation of humanistic artifacts. From this it follows that humanities must consider the extent and characteristics of a hermeneutics that takes digitality and computationality into consideration. Current practice shows, if it was not already self-evident, that conventional hermeneutics in its form of "post-algorithmic" interpretation takes up a large and undeniable part. At the same time, as we apply algorithms, models, and quantification, there arises an urgent need to understand the effect of these analytic methods on our hermeneutics. We have seen that the design of analytic methods is not free of its own hermeneutics. The effects and ramifications of these implicit hermeneutics on humanistic interpretation and reasoning are nevertheless unclear, poorly understood, and hardly studied. To understand these effects more fully – that is, to understand the hermeneutics of algorithmic and quantified approaches – we need a constructive and intimate dialog with the domains of computer science and software design. We cannot simply face outward after the algorithmic fact and interpret its results without implicitly but unconscientiously being a proxy to its hermeneutics. The profound effects of the digital on human culture and the humanities demands that we fully grasp its potential for hermeneutics.

Chapter 3

The Case of the Bold Button: Social Shaping of Technology and the Digital Scholarly Edition[1]

> "First, let us observe two things missing from almost all electronic scholarly editions made to this point. The first missing aspect is that up to now, almost without exception, no scholarly electronic edition has presented material which could not have been presented in book form, nor indeed presented this material in a manner significantly different from that which could have been managed in print."

These are words by Peter Robinson, who spoke and wrote them in 2004 (Robinson 2004). I think little has changed in over a decade since and the observation still more or less holds. At the time, Robinson argued vehemently for digital scholarly editions that would move decisively beyond the realm of the possibilities of print publication. He was – and is – by no means the only one that has been advocating for such a shift. In fact, many have wondered how the digital medium, or the virtual environment, would change the nature and appearance of the scholarly edition. For that matter, grand

[1] This chapter appeared before as Van Zundert, Joris J. 2016. "The Case of the Bold Button: Social Shaping of Technology and the Digital Scholarly Edition." *Digital Scholarship in the Humanities* 31 (4): 898–910. https://doi.org/10.1093/llc/fqw012. It was updated to include a more expanded definition and explanation of the concepts of "graph" and "knowledge graph". A few sentences were added to clarify my use of "heuristics", and the meaning of "minimal" and "maximal edition".

perspectives on paradigmatic change due to medium change are not unique to textual scholarship. The introduction of a new medium or technology has always inspired great debate between advocates and antagonists of the next big thing. Self-proclaimed supporters of digital media usually advocate revolutionary changes. In the case of textual scholarship, for example, one may hear it proclaimed that the book is dead; good riddance, the advocates for "The Next Big Thing" (Bod 2013b:8) judge, for it was a clumsy, static, institutionally bounded, difficult to use, and outdated interface. Give way to open access, process orientation, dynamic interfaces, intuitive interaction, fluid text, social editing, etc. (cf., for instance, Siemens et al. 2012). With similar and undaunted zeal, Luddites lament the waning of solid scholarly practice: concentration span, close reading, philological interpretation, editorial practice, and convention (Fish 2011) – all sacrificed to the "Bitch goddess, QUANTIFICATION" (sic) as Bridenbaugh once put it (Bridenbaugh 1963). Or, for a more recent example in the Dutch literary and linguistics theatre, consider professor Rens Bod proclaiming the end of Humanities 1.0 (Bod 2013b), and Ph.D. student Marieke Winkler sincerely questioning that (Winkler 2013).

The screaming and kicking of Luddites aside the proponents of change do not seem really to get what they want. After many years of development of digital technology, the book is as alive as it ever was. We scarcely find digital editions, scholarly or otherwise, resembling the advanced models of dynamic, fluid, collaborative, and social texts such as those proposed by McGann (2010), Drucker (Lunenfeld et al. 2012:36), Shillingsburg (Thiruvathukal, Jones, and Shillingsburg 2010), Robinson (2004), Van Hulle (2010), Siemens (Siemens et al. 2012), and myself (Boot and van Zundert 2011). E-books are certainly impacting the market (AAP 2010; Cain Miller and Bosman 2011), but e-books are pure digital metaphors of the print book. Digital scholarly editions hardly have any impact (Porter 2013). What is more interesting is that digital scholarly editions are a far cry from what many expected them to be. We could suppose that this state of affairs is due to a lack of knowledge, skills, and technology support as has been indeed suggested before (cf. Courant et al. 2006). And it is probably true there are severe problems of teaching and training in our field, given that master and Ph.D. programs truly oriented on the digital humanities are only lately coming into existence. Yet, I think there might be more to the matter.

Maybe we need to answer to Borgman's call (2009): "Why is no one following digital humanities scholars around to understand their practices, in the way that scientists have been studied for the last several decades?" What do we see if we step back for a while from our work as textual scholars and digital humanities researchers and look at what is happening from the social sciences, in particular of Science and Technology Studies? Science and Technology Studies suggest to study technology development in its social context. In the past few years, I have studied the creation and development of the digital scholarly edition within the laboratory-like setting in the Huygens Institute for the History of the Netherlands. Here we find a relatively large – for humanities contexts in any case – IT Research and Development (R&D) group of on average sixteen persons working together with about sixty historians, textual scholars, and digital archivists. The research context consists of a dozen senior researchers, a similar amount of non-senior and associate researchers, a similar amount of Ph.D. candidates with various contracts ranging from predominantly full-time added staff to volunteer workers, and of course non-IT R&D supporting staff. As in many other contexts (cf. Nowviskie 2012) the relationship between the IT R&D group and scientific staff is some matter of internal debate in the institute. In part, the role of the IT R&D staff is in support, in part it is collaborative at the research level.

The adoption and application of technology is as much a social as it is a technical process. These processes are inevitably intertwined: technology does not determine but operates within and is operated upon in a complex social field (Bijker et al. 1987). The manifestation of such intertwined processes is directly visible in the field of digital humanities and in the development of the digital scholarly edition. Of course, the digital scholarly edition is a digital artifact brought to life in a context of heavy interaction between a highly digital technological community (computer scientist, software engineers and digital humanists) and a far less digital technology based community (textual scholars). This intricate and intensive interaction is a daily practice at the Huygens Institute for the History of the Netherlands. One of my tasks has been to guide the interaction between IT R&D, documentary editors, textual scholars, and researchers of literature and history, and to facilitate the ongoing methodological discussion between these cultures. I have had the privilege to study these processes from many angles: methodology, technology, model, role, audience, development, and so on.

As has happened in many similar research contexts, a transcription tool was developed at the institute to support the basic work of turning non-OCR-able texts from early printed works and medieval and modern manuscripts into digital machine-processable counterparts. The development of this tool, eLaborate[2], was based on a strategy of encapsulating and hiding XML markup – to be transformed to TEI encoding behind the scenes – with a graphical interface. In this way, the tool was meant to present minimal barriers to transcribers who came in a variety of levels of expertise on encoding. This indeed resulted in successful participation of significant numbers of volunteers unskilled in XML over a large set of projects. Also the GUI encapsulation of XML technicalities facilitated greatly the focus on community and project management (Beaulieu, Van Dalen-Oskam, and Van Zundert 2012).

Here I am not so much interested in the features or particulars of eLaborate. Instead I want to focus on one particular researcher-developer interaction I witnessed that, I think, stands as an example of a general and strong tendency in the scholarly community at large. The usability principle behind eLaborate is that any encoding or markup is treated as an annotation on arbitrary regions within the text. To this end, when a user has selected a certain region in the text with the mouse, a pop-up dialog appears allowing the user to enter annotative tags, comments, etc. The interface thus closely mimics a concept – using a highlighter and pencil to create annotations – that is known and tangible to anyone who has a basic experience in working with scholarly texts. The clear downside of this principle – if dogmatically applied – is that a user is faced with an enormous number of click-and-point-and-type annotation operations. Especially in cases of seemingly insignificant but frequent markup, such as with the indication of bold face print, this approach can strike the user as tediously and needlessly pedantic. It should be noted, however, that regarding the feelings of tediousness a distinction is likely to

[2] Cf. https://www.elaborate.huygens.knaw.nl

be made between senior scholars as transcribers and non-academic volunteer "crowd sourcees". Evidence from the projects, if mainly anecdotal, suggests that volunteer transcribers in fact may attach hundreds of tiny and similar annotations without complaint, but that the senior researcher will feel put at odds with her experience and practices when invited to do so. In any event, the upshot of this usability agony was a recurring and strong push in the user community to have a button labeled "bold" – in fact to have several such buttons for italics, underline, and other common very frequently appearing properties of text – lowering the volume of tedious annotation. To this day I remain convinced that we should not have implemented that button as we did.

The foundation for my conviction is that these buttons violate the rationale for XML over HTML, namely the strict and intentional separation of representational and semantic information. The most common interpretation of boldface type is that it is a material manifestation of the concept of emphasis. Even this is not universal – many other concepts may also be expressed by the use of boldface type. Thus, the provision of a button to record that some text is in boldface type introduces inevitable ambiguity in a descriptive system. At a later point in time there is no unambiguous way to tell what the function of the bold print was: it arbitrarily covers many intentions without delineating which of several possible textual concepts might apply.

However, more important for my argument here is that the implementation of this simple button reveals how technology is indeed shaped through its social context. The intent of eLaborate's approach was paradigmatic: its purpose was to allow editors of text to change from a representational paradigm to a semantic paradigm. We could have done this by forcing our users to become competent XML authors. Our users judged XML tedious and complicated, however, and complexity is a well-known "fail factor" working against the adoption of any new technology (Rogers 1983). Thus, to guide our users gently into a new paradigm, we had to create an interface that offered a clear and substantial advantage over existing technology and at the same time did not seem overly complex. The annotation "highlighter" pop-up seemed a good solution that tried to balance innovation with ease of use and some compatibility with a known paradigm. However, the annotation pop-up led to a tedious routine that in reality severely curtailed the ease of use. When ease of use is compromised to such an extent new possibilities inherent in a technology do not lead to a change of routine to accommodate the technology, and thus an adoption of the new paradigm does not occur. Instead, the perceived constraints lead to a change in the technology (Leonardi 2011). This is exactly what happened in the interaction between developers, users, researchers, and technology in the case of eLaborate. A bold button was introduced to remedy usability constraints: social shaping of technology at work.

As an unintended consequence – as Robert Merton would have it – of this social shaping of eLaborate the paradigmatic intent of the innovation was now black boxed. This is not meant in the sense of Latour's definition that defines a black box according to general acceptance of the correctness of the inner mechanism (Latour 1987), but in the sense that the innovative aspect of the new paradigm was now completely unobservable and thus effectively unknowable to its intended audience. The unobservability of such a black-box model is also a known "fail factor" for innovation (Marinova and Phillimore 2003; Rogers 1983). I my experience I found interfaces to often have this unintended and usually unrecognized effect, and it is a problem that particularly affects graphical interfaces. A graphical user interface suggests a transparency of model and paradigm that is not truly there – in fact the graphical interface is as much an opaque barrier to the internal paradigm of a system as it is a means of engaging with that very system. Analogous to Robinson (2013b) and others, I would argue that software interfaces, such as the interfaces to digital text editions, are an intellectual argument about the internal model of a system rather than a neutral communication of that model to any user. Vice versa, when the interface undergoes social shaping as a result of the interaction between developer and user/researcher, that is also an expression by that user of an intellectual argument about the model.

In the case of the bold button, the user has not merely molded convenience into the interface. What also happened was that the intended paradigm – that of semantically oriented XML – was expressed in a paradigm which was more familiar to most users, namely that of representationally oriented HTML. But this effectively prevented the user from engaging with and getting to know the new paradigm, or at least a part of it. The bold button hid a class of semantically expressive potential behind a single representational "wrapper". As an extension of the Meno paradox (Nickles 2003), not only were the users unable to negotiate new knowledge, they had shaped the technology in a way that made it now impossible to engage at all with the new paradigm. User-centered design had led to the users shaping new technology so that it was congruent with the paradigm they were familiar with. The new was expressed in the ways of the old, but it also turned into something inaccessible and irrelevant. This unintended effect of an intended paradigm being encapsulated and effectively hidden by a more familiar paradigm is caused by what I will call paradigmatic regression: the social shaping of a technological interface such that it can no longer express essential properties of an intended paradigm. The pivotal error that was made with the introduction of the "bold button" was that the button did not express the digital paradigm. Instead, we did exactly the opposite: we facilitated the scholarly users' regression toward the paradigm of the book metaphor known to them. Thereby we confirmed that nothing had changed, that print convention was still the paradigm to use. As proponents of digital scholarship, we may tend to think we are free from this sort of paradigmatic regression. But we are not. Most if not all digital scholarly editions are still solidly rooted in book metaphors and print conventions, and I think it is exactly because of this silent regression. A brief and selective history of humanities computing may be telltale.

The beginnings of humanities computing and the development of the digital scholarly edition are usually dated to 1949 with the seminal work of Father Busa (Hockey 2004). Roberto Busa demonstrated the first practical applications of computational text processing by automating the tasks of indexing and context retrieval. However, the result was presented in a form already well known to scholarly editing: a fifty-six-volume print publication concordance. The computational aspect was used simply to automate and scale a tedious and error-prone editorial task. The utility and sense of this work goes without question. What interests me here, however, is that the automation was geared toward reiterating on a larger scale a scholarly task that was in essence well known and rehearsed; computational power was harnessed to produce an instrument well within the confines of the existing paradigm of print text and its scholarly applications.

The advent of the database and later the relational database prompted the curation and publication of several catalogs and indices of textual metadata, as well as the first repositories for digital text. This was of course a major enhancement of the capacity for discovery of texts and related metadata. Databases allowed for efficient and convenient discovery of text through the use of matching selection queries. Scholars such as Jerome McGann, Peter Robinson, Dino Buzzetti, Manfred Thaller, and others began to envision different forms of engagement with text made possible due to the availability of full-text repositories and metadata. Despite all this, the database did not change the essential way scholars engaged with the actual texts. Even if, for instance, Buzzetti and Thaller argued that a digital edition's "liability to processing" is the essential feature that sets it apart from conventional editions (Buzzetti and Rehbein 1998), texts were still perceived predominantly as intentionally ordered strings of words for human interpretation. Thus, notwithstanding ideas on how to engage with text in new ways separate from the reading, commentary, and interpretation that has traditionally been handled by humans, the digital scholarly editions produced in the last part of the twentieth century have again presented text to us essentially as a digitized book.

According to Hockey (2004), in the early to mid-1990s a great deal of interest and discussion arose in the scholarly community concerning what an electronic edition might look like. However, with the "notable exception of work carried out by Peter Robinson", few of these publications were realized in an actual implementation. Once "theory had to be put into practice and projects were faced with the laborious work of entering and marking up text and developing software, attention began to turn elsewhere". As with the bold button example, we find that a new technology turned out to provide too little practical facility to lead to successful innovation. Yet there is more to the matter.

The "Next Big Thing" of the last decade of the twentieth century was the World Wide Web, founded on the technologies of the Internet and hypertext. As Landow has pointed out, "computer hypertext – text composed of blocks of words (or images) linked electronically by multiple paths, chains, or trails in an open-ended, perpetually unfinished textuality described by the terms link, node, network, web, and path" precisely matches Roland Barthes' ideal textuality (Landow 2006). If we need to point to a single moment and opportunity in history when the very fabric of a new technology was made suitable to a scholarly community for the expression of relations and structures, not just within single texts but especially between texts, it was the moment of the invention of hypertext. That the opportunity arose cannot have been surprising, as the essential mechanism of hypertext – the hyperlink – was the technological implementation of a long-standing idea that knowledge and information are interlinked. Already pioneers such as Paul Otlet in the early twentieth century could contemplate information systems that would link knowledge in the form of formalized multidimensional relations between documents (Rayward 1994). What is actually rather surprising is that such long-standing epistemological knowledge about the relation of different chunks of information within documents and congruent ideas from post-structuralist literary criticism such as Kristevas intertextual references (Mitra 1999) found so little expression in digital scholarly editions. The expressive power of that single pivotal element of the original HTML 1.0 specification, the A element with its invaluable HREF property, implemented by Tim Berners-Lee and itself an echo of Theodor Nelson's ideas of transclusion (Nelson 1995), should have reverberated within the scholarly community. The Hypertext Markup Language (HTML) enables anyone to link one HTML document to another by inserting the anchor element and pointing its HREF attribute to the web address of the other document. So when you see a link on a webpage (e.g.the blue headings on search results in Google) then in the source code of that page the link is encoded similar to this example:

```
Further work has been pursued by researchers of the
  <a href="https://www.huygens.knaw.nl/">
    Huygens Institute
  </a>
.
```

On a webpage that <a> element ensures the words "Huygens Institute" will be a clickable link, and clicking the link will forward you to the URL (i.e. web address) encoded in the href attribute of that same element.

This simple mechanism should have created ample opportunity for editors to give expression to the linked and intertwined natures of cultures of text, literary criticism, and (digital) textual materiality that go to the heart of the field (Van Mierlo 2006). The hyperlink created a native digital expression for the act of referencing, an expression of knowledge very much at the core of textual description, interpretation, and criticism. Thus, here was a unique opportunity to change from a paradigm of print publication to a paradigm of interconnected texts expressing knowledge.

The scholarly editing community, however, adopted the "markup" rather than the "hyper" part of the hypertext markup language, by developing Goldfarb's SGML eventually into the TEI-XML descriptive standard (Goldfarb 1996; Renear 2004). At the time, these dialects of markup technology were used primarily to mark up texts as they are represented in books – the fact that I do not think anyone has but flippantly suggested marking up Web pages in TEI-XML may stand to prove the point. The scholarly community predominantly turned hypertext markup into a descriptive model of the book, and we have produced digital book metaphors as digital scholarly editions ever since. As with the bold button, a new technology was not explored but rather encapsulated by a known paradigm. The hyperlink was meant not to be a descriptive tool, but to link information in different documents. Yet its foremost use in scholarly editing has been to link contents, chapter headings, and indices to pages in self-contained digital editions. Roberto Busa had "a vision and imagination that reach beyond the horizons of many of the current generation of practitioners who have been brought up with the Internet". He imagined scholarly editions on the Internet combined with analysis tools (Hockey 2004), a horizon that has been reiterated by many (cf., for instance, Buzzetti 2009). However, digital editions developed in a completely different direction. The processing involved is mostly aimed at rendering the text for consumption by human readers. To defy the intent of the hyperlink has been in my view among the most remarkable feats of paradigmatic regression in the textual scholarship community. One can wonder though whether this is a bad thing. If we accept the bilateral dynamic between audience and innovation, then why would we care when some innovations do not succeed? If the book metaphor paradigm suffices for our needs, does this not indeed suffice?

To answer this question we must ask: to whose needs do digital scholarly editions actually cater? Given the designation, they should cater to scholars and researchers, but do they? The latest developments in digital scholarly editing are linked to the possibilities created for Computer-Supported Cooperative Work (CSCW) – a term that was coined by the IBM research group headed by Greif (1988) – by networked computing, the Internet, and the rise in computer literacy. Essentially CSCW is a label that can be put on any collaborative activity that is supported by Web or Web 2.0 means. Crowdsourcing as a means of dividing large work-loads has been around for a while and has been a specific implementation of CSCW ever since Web 1.0 technologies turned into Web 2.0 technologies. Many have proclaimed crowdsourcing to be the advent of the social edition – most prominently Ray Siemens (Siemens et al. 2012) – which redefines the editor's role to be that of a team leader concerned with proper workflow, quality control, and overseeing managerial and funding aspects (Sahle 2013), whereas concrete editorial tasks are delegated to social communities formed around specific texts. Questions have been raised about the actual effectiveness of crowd sourcing (Causer, Tonra, and Wallace 2012). But more importantly, recent studies show that the old rule of thumb of the collaborative Internet – that 10% of the workforce provides 90% of the labor (cf. Brumfield 2012; Brumfield, Klevan, and Vershbow 2012) – still holds for any open collaborative project, implying that many crowdsourced editions are not in fact truly social. Moreover, when Peter Robinson said "All readers may become editors too", he was not simply referring to a cheap labor force for source transcription, to be conveniently discarded the moment a transcription phase is done (Robinson 2004). Instead, like Ray Siemens proposed, he envisioned a "social edition" that embodies the ideas of open notebook science (cf. Shaw, Buckland, and Golden 2013) and renders all aspects of the editorial process – e.g. annotation, commentary, and interpretation – open to public engagement (Siemens et al. 2012). But we in the scholarly community are not at all at ease with letting go of our presumption that scholarly editing is a highly skilled practice that does not provide for easy delegation of tasks. It is challenging to truly consider the extent to which we can open up the scholarly process of creating a digital edition to leave the tedious tasks typically associated with high quality scholarly inference to the wisdom of the crowds – in the case of literary analysis, this often includes, for instance, the painstaking tracing of names, annotation of plot, and clarification of meaning. In current practice, however, the digital scholarly editorial tasks beyond the transcription phase remain reserved either for the single authoritative author or for a small group of qualified editors. In this way, most scholarly digital editions adhere to an authoritative publication paradigm. We use big all-encompassing words like "social", "open", and "community", but in fact we are again regressing to authoritative processes that remain well within the paradigm of the print edition. Although on the verge of being harsh, it is nevertheless fair to state that digital scholarly editions cater to the needs of the scholarly editors, not to users and researchers as knowledge producers.

Along another tangent: Edward Vanhoutte (2011) pointed out the possibilities of targeting different audiences with different visualizations of the same edited digital text resources. So-called "minimal editions" – essentially filtering down all resources to provide a polished and uncomplicated reading text – could cater to a broader audience while "maximal editions" would cater to the use of scholarly researchers, providing all scholarly explanations, variations, annotations, and so forth. Several digital scholarly editions do show signs of this sort of differentiation. We can point to the Van Gogh Letters (Jansen, Luijten, and Bakker 2009) as something of a midpoint between the minimal and maximal edition. The Samuel Beckett Digital Manuscript Project (Van Hulle and Nixon 2011) and the pre-production version of the Digital Faust Edition (Brüning, Henzel, and Pravida 2013), that I was allowed to peruse while it was in development, certainly should qualify as maximal editions. However, these digital scholarly editions again reiterate in their GUIs the metaphor of the 'read-only' book.

Only very few digital scholarly editions do provide what I think is paramount for true interaction with editions or scholarly text resources: the capacity to negotiate the edition and its text as data over Web serviced Application Programming Interfaces (APIs). APIs allow for computer-to-computer negotiation of texts, opening them up to algorithmic processing and reuse. My primary reason for arguing that we need digital scholarly editions as API accessible texts is not, as some may expect, to enable quantified computational approaches such as those that Matthew Jockers and Franco Moretti have presented (Jockers 2013; Moretti 2007), or the stylometric analysis desired by many others (Van Dalen-Oskam and Van

Zundert 2007; Kestemont 2012). It is highly useful and convenient to have the text of scholarly editions available as an open Web service, so that my computational colleagues and I can do our principal component analyses, bootstrap consensus trees, clustering analyses, and any other analysis that can possibly be envisioned.

But there is another reason, in my view more important yet overlooked, to consider anchoring digital scholarly editions on a data model that is not oriented around a book metaphor. This motivation derives from the growing and increasingly unsettling gap I find between the close reading of scholars using conventional hermeneutic approaches and the "big data" driven distant reading supported by probabilistic approaches – a discrepancy which is also signaled by others (e.g. Capurro 2010). On the one hand, we see a conventional scholarly approach in which texts are mindfully and meticulously produced, detailed, and interpreted. On the other hand, we find a deterministic and probabilistic approach that focuses on large-scale data analysis and which is, through its statistical aspect, reductive in nature. To the hermeneutic scholar, distant reading approaches are therefore "lossy", prone to discarding some of the substance, and quite incapable of capturing essential hermeneutic knowledge (cf. Ramsay 2011c). It is often the statistical outliers and not just patterns of similarities that are telltale to textual scholars and historian in their hermeneutic explorations. At present there is no model connecting these worlds of close and distant reading. Rather, the distance between them is growing, which threatens not only to set the scholarly community of textual and literary studies against itself, but also to waste the opportunity for a true and meaningful advance in our capabilities for computational-based humanities research.

If we are to close this gap, we need a model for digital text that allows for both hermeneutic and statistical approaches so that these approaches can truly inform each other. To this end we need to revisit and reconsider how we anchor digital editions on the hypertext model. The slavish adherence to the book metaphor, even in XML form, will not take us into a realm where texts and editions are published as online APIs for processing by computational means. Yet, also models of quantification fall short as they are narrowly defined for statistical methodology. Because such models are not data models, they do nothing as to expressing description, encoding, or annotation. We are in need of a model that actually provides for all of the above. That is, a model that provides for the capturing, encoding, and annotating of a text and also for processing the edited or raw resource to enable analyses by both conventional hermeneutics and quantified approaches. Lastly, this model must be recursive: it must be able to capture all resulting information from an analysis and add that information into the model itself. Only then new knowledge gained from the model can be used 'natively' for a next cycle of both qualitative and quantitative analysis. Such a model captures all editorial and research aspects and outputs of scholarly activity in an encompassing lifecycle. But even more important: only such a model provides for a way to bridge the widening gap that is coming into existence between the hermeneutic tradition and new quantified means. Computational method can do far more than just counting, averaging, and comparing histograms. But currently computational approaches ignore many of the properties of text and textual materiality that are important to hermeneutic engagement. Current quantified approaches lack therefore the ability to model and computationally process the close reading aspects of text engagement.

Thus what we lack is something we could call tongue in cheek near distant or near close reading. More formally and in line with current debate, I think we should qualify what we lack as an enabler of computational heuristics for capta (Drucker 2011). There is no computer language that specifically supports the type of abductive reasoning that textual scholars and literary researchers do. Their heuristic is a scholarly adequate but not very formally qualified set of rules and activities underpinned by knowledge and experience from reading, studying, and discourse. These heuristics are applied to combine both evidence and plausible assumption to infer interpretations of by definition situated – that is, context dependent – cultural artifacts. It is thus a method to construct interpretation, the result of which is what Drucker calls "capta". Arguably either 'near close reading' or 'near distant reading' both capture in their own ambiguity exactly the properties of textual scholarly data, knowledge and method that quantified approaches tend to overlook: extremity of sparseness, inconsistency, vagueness, ambiguity, multi-interpretability, and uncertainty. There is no readily available means for such qualitative computing. Qualitative modeling and computing are still highly explorative fields (cf. Forbus 2008), and yet, abilities to compute and reason over qualitative data are coming into existence. As the creators and providers of the raw materials that such qualitative computational approaches should operate on, editors of digital scholarly editions should consider how text as data is to be provided.

Knowledge graphs are, I think, extremely well suited for this. Knowledge graphs represent our objects, concepts, and properties of interest and their relations in a network. They consist of nodes that usually represent objects or concepts. These nodes are connected by lines, called edges, that usually represent relations. As a matter of fact, the constructor of a graph is completely free to determine the meaning of edges and nodes. If he or she chooses so the edges might be the concepts and the nodes types of relations. One can imagine that all tokens in a text might be individual nodes and that the edges determine the linear order in which they appear in the text. Another graph might represent characters in a novel and the edges might model the relations ("friend", "sister", etc.) between them. In essence knowledge graphs are a formalization to express predicate semantics, two nodes represent object and subject, the edge indicates the predicate (figure 3.1).

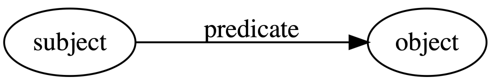


Figure 3.1: The predicate principle of graphs.

Graphs are a most generalized class of construct of which ontologies and thesauri are subclasses with more specific constraints. In their most generalized form graphs are stupefyingly easy to express in computer readable form using, for instance, the DOT format or language (Hayes-Sheen 2017):

```
graph {
  mary -- rose[label="friends"];
}
```

This example results in the graph of figure 3.2, when processed with a suitable program such as Graphviz[3]. The salient point is not that this results in nice visualizations, but that this simple formalization serves as a facile interface between human expression of knowledge and information that can be processed by the computer and be reasoned with by algorithms. A very baseline example of this would be to infer who is a friend of a friend in a more complex network.

Figure 3.2: A most basic graph, showing two nodes and one edge.

Graphs are not new to us, nor to our field. The World Wide Web is a graph, a network of nodes and edges connecting information. In a sense, every digital scholarly edition put online has in fact been made part of a graph therefore. In recent years, graphs have found various more explicit applications also in the field of digital humanities, most notably as a data model for describing textual variation between different witnesses of the same text (Schmidt and Colomb 2009). The properties of the graph model, however, allow it to be a generic model capturing the information tied to a digital scholarly edition on all conceivable levels of granularity. Two examples may show this potential conceptually.

The first example was kindly explained to me by Moritz Wissenbach who at the time was a technical lead working on the Faust digital edition from Würzburg university. Imagine a knowledge graph as a network with nodes and edges. In this hypothetical graph, we designate three nodes to represent texts A, B, and C. An interface to the graph allows us to add edges and nodes to this network. What is essential here is that the underlying model is a graph, the graphical display may take many forms but need not necessarily be a visual network itself. Suppose now a textual scholar X states that text A was conceived before text C. This statement can be represented as a directed relational edge (or predicate if you like) 'precedes' between A and C as depicted in figure 3.3. Now assume another researcher Y at another point in time, and

[3]http://www.graphviz.org/

not necessarily even knowing anything about text A, independently of researcher X, concludes that text B was conceived after text C. This statement can be captured by putting an edge 'precedes' between C and B. The tiny graph as depicted in figure 3.4 now holds the accumulated knowledge. However, note that the combination of independent observations now adds up to more than just the sum of its parts, for "traversing", "walking", "reasoning over", or "computing over" the graph – all these terms essentially express the same operation of computationally inferring knowledge from the graph – gives us the added knowledge that A must have preceded B.

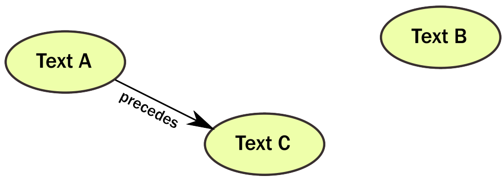


Figure 3.3: Nodes in a conceptual knowledge graph.

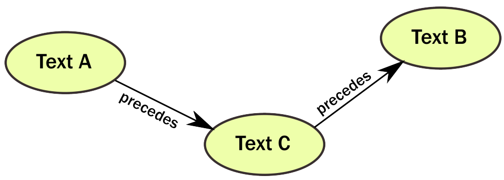


Figure 3.4: Edges multiply knowledge.

The second example is taken from CollateX, which is a tool to automatically collate variant texts (cf. http://collatex.net/). The result of such comparisons can be stored as graphs, e.g. figure 3.5. Such graphs cannot be said to be quantified, they express rather the qualitative word variance between texts. But the application of the graph stretches wider. As in the previous example, we can add statements (knowledge) about this text to the graph by adding nodes and edges. The example in figure 3.6 shows two statements made by superseding nodes on partly overlapping regions of the text. They express in a hypothetical fashion how these regions should look for a reader of an

EPUB publication of the text to be read on an eReader. Note how overlap, a well-discussed problem for hierarchical models (Sperberg-McQueen 2002), is not relevant to such a non-two-dimensional graph model.

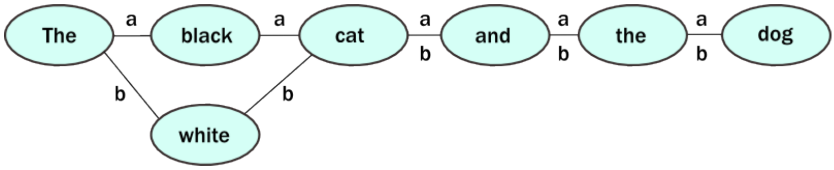


Figure 3.5: Conceptual knowledge graph representing textual variation in two texts a and b.

It should be carefully pointed out that knowledge graphs as a model are not to be equated with the currently popular ideas on semantic Web and RDF. RDF can necessarily only be a static representation of a certain state of such graphs that rather would be dynamic representations of changeable knowledge. To solve this initiatives such as the Open Annotation Collaboration are proposing extensions to the World Wide Web and Semantic Web models to support annotation of linked data including temporal "aware" annotations (Haslhofer et al. 2011). It is out of scope for this chapter to examine whether such models would provide for the needed reciprocality and dynamics for graph model-based digital scholarly editions. As the Web in its current form is not real-time read/write enabled, it is hard to imagine though how it would provide for such highly dynamic webs of knowledge interaction. The relation between RDF/Semantic Web and graph models is somewhat analogous to the relation between TEI and XML. A TEI conformant XML document is a singular instantiation of (a part of) the TEI model. The TEI model itself however is represented by the dynamic set of guidelines defined for the description of text and document structures.

Knowledge graphs can grow dauntingly complex very quickly, as may be inferred from figure 3.5. Because such complexity also poses a problem for querying and performance on the computer science side of things, we have not seen wide application of graphs until now – let alone as a model for humanities data. However, meanwhile knowledge graphs in the same fashion as shown in these tiny examples back the social network applications of,

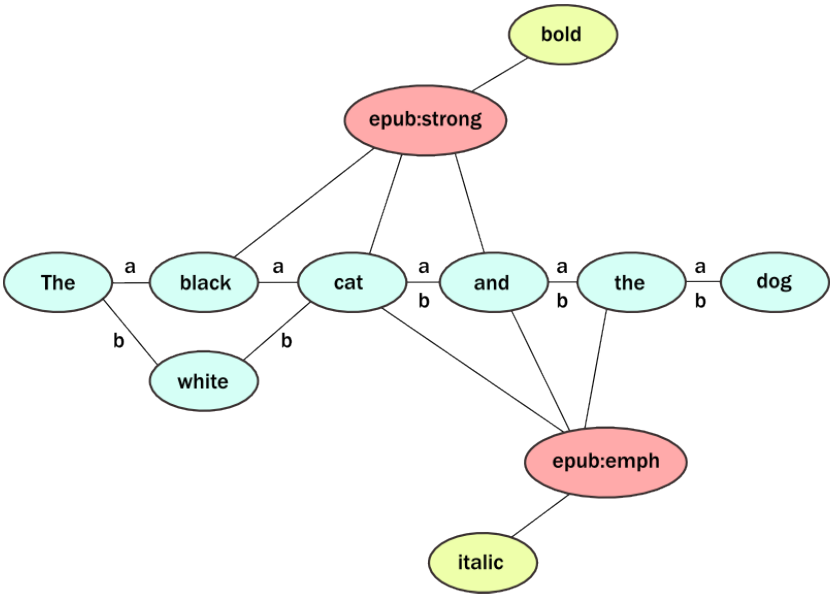


Figure 3.6: Overlapping semantic and representational knowledge added to the graph of figure 3.3.

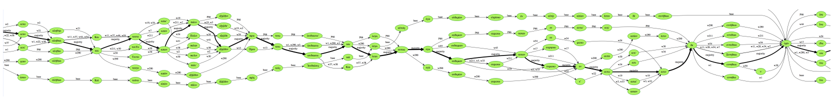


Figure 3.7: A graph representing a bible verse in various redactions.

for instance, companies like Facebook and Google. Graph databases[4] like Neo4j, ArangoDB, and AllegroGraph have made application-level models feasible. This paves the way toward exploring the potential of graphs for expressing the information and knowledge represented in digital scholarly editions. In reality when putting text and editions on a graph, as users we may not experience them as graphs, but rather as any visualization or data representation we want to derive from the graphs. By footing such representations and visualizations on a graph model, we provide an underlying truly generic and interoperable means for representing, editing, annotating, and visualizing text, its relations, its multi-perspectivity, and its materiality in digital scholarly editions. At the same time and through the same data model we provide a means for qualitative and quantitative computing over the information contained in the graphs representing our editions. Thus, with a graph model, we provide a more expressive data model for digital scholarly editions, allowing for the modeling and computation of both statistical and hermeneutic approaches.

Providing a digital scholarly edition with the backbone of a network graph would mean anchoring text on a fundamentally different model than that of the current prevalent digital book metaphor. All digital book metaphors are until now essentially closed off inconvenient mixtures of multiple page and string oriented hierarchical models. What we cannot achieve through the book paradigm is walking the various alternatives of the graph that expresses interpretations and knowledge about the document in consideration. That is, we cannot algorithmically get at and process the text with all its annotations, comments, and additional information on authorship, materiality, interpretation, etc. The reason for this is that the book paradigm keeps us locked in and focused on a finite representational state of the text: it is oriented toward closing down the text. In contrast, graph models provide an elegant open way to connect information to the text in an infinite extensible fashion. Whether machine negotiated or by human interpretation, new information can be attached to any particular item in the graph in the same way, thus becoming information that can be processed by both scholar and algorithm. Thus, the essential difference is that the same model can cater to capturing hermeneutic inference and computational analysis results. But

[4]https://en.wikipedia.org/wiki/Graph_database

we will only successfully explore that potential if we quit the social habit of shaping back new models into old paradigms.

Chapter 4

Barely Beyond the Book?[1]

> "There is nothing deterministic about the Internet." – David Lowery, frontman of Camper Van Beethoven and lecturer at Terry College of Business, University of Georgia.
>
> (Timberg 2013)

This is a story about the methodological interaction between two scientific fields, that of textual scholarship and that of computer science. The names of the fields, however, only imprecisely delineate the permeable boundaries between research domains where methodologies interact – for obviously the world is much more fluid than such nouns suggest (Simon 1995:234). The interactions of interests are much more complex than the simplified image of a dynamic whereby one field donates a methodology to another. Rather than trying to reflect on the current state and the future potential of the digital scholarly edition from well inside the field of textual scholarship, let us approach the topic from the perspective of the multidisciplinary methodological interaction that has arisen to support the theoretical and practical development of the digital scholarly edition over the recent years. Textual scholarship in its digital fashion belongs to the broader field of digital humanities, itself a field built on interdisciplinarity, where many skills and theories of the

[1] A previous version of this chapter was published as Van Zundert, Joris J. 2016. "Barely Beyond the Book?" In *Digital Scholarly Editing: Theories and Practices*, edited by Matthew James Driscoll and Elena Pierazzo, 83–106. Cambridge (UK): Open Book Publishers. http://www.openbookpublishers.com/reader/483. The observations on methodological changes regarding the classic "index" in the case of the eLaborate project were expanded for clarity.

realms of computer technology and those of scholarship intersect, and thus where many new interfaces and interactions arise between those skills and the fields they are tied to (Hockey 2004). This is where digital humanities acquires its innovative power, or at least the promise of that power.

That innovative power, however, can be both exciting and confusing. The point where disciplines intersect is not a space for the calm, cool and collected exchange of technical and methodological knowledge. Rather, it is a place where the inherent social aspects of science and research are brought markedly into the foreground (Latour 1987). Take for example Jan Christoph Meister's description of the "lamented conflict between 'computationalists' and 'humanists' ". This conflict, Meister (2013) states, "arises as soon as we become afraid of our own courage and shy away from jumping across these two fault lines. Let's cut through that fear. The task remains [...] to 'become capable of both – the metaphor and the formula, the verse and the calculus [...].' That's a borderline experience, no doubt, and those who prefer to pitch their tent in the comfortable centre of either laager don't run the risk of questioning their own philosophical, epistemological and ethical identity as easily." Meister's word use is notably emotive ("afraid", "fear", "courage") and at the same time vividly touches on the impact of the social dimension ("conflict", "borderline experience", "risk") of the epistemological interaction that is expressed. As Christine Borgman has suggested this is a situation where it can be useful, with respect to the design of scholarly infrastructure, to take these interactions and the behavior connected to them as the objects of study (Borgman 2009). Let us do exactly that here. Taking the digital scholarly edition as a part of the scholarly infrastructure for textual scholarship, we can try to infer what the historical interactions between textual scholarship and computer science tell us about the current state and development of the digital scholarly edition.

The field of science and technology studies (STS) offers a useful frame for critical study and reflection on what occurs at the interfaces of the various research fields within digital scholarship. When these fields intersect, it is not simply a question of objective interactions concerning technology and methodology; rather, these interfaces are also the sites of social processes that guide and steer the methodological interaction. Within STS such processes are often referred to as the social shaping of technology – that is, the mutual interplay between technology, its developers or champions, and the users of that technology. It is this interplay that changes properties and application of the technology at hand. For example, such interplay is very prominent in software development, in which development iterations and lifecycles are a clear expression of the interaction between builders and users as they shape software until the users' requirements are satisfied (cf. e.g. Lee and Xia 2010).

In the previous chapter I argued that social shaping of technology can lead to 'paradigmatic regression'. These are acts of shaping that translate an expression of the paradigm of the new technology into an expression of a paradigm that is already known to the user. Resistance to new technologies, where the use or sophistication of the new technology is denied, can of course be a motivator of paradigmatic regression (Levallois, Steinmetz, and Wouters 2013). Not all regressions are necessarily motivated by conservatism or resistance, however. But even when users do embrace a new technology, the act of its social shaping may create a paradigmatic regression effect. An example of this effect can often be found when a metaphor is used in a graphical user interface (GUI). GUI metaphors are used to convey the processes or data underlying a particular piece of software in a manner that is meaningful or intelligible for human users. In order to help the user understand a new target domain or a new paradigm, it is expressed by way of a conceptual domain or a paradigm that is already known to the user. An obvious example is the metaphor of the desktop, which was used to communicate the functions of a personal computer (PC) to as broad an audience as possible (cf. Baecker et al. 1995). The only trouble is that such metaphors are necessarily incomplete as they conceal both the good and the bad of the deeper computational model. Inconsistencies in the model are hidden by a metaphor that suggests completeness to the user. Equally, metaphors hide useful functions and possibilities of the model that are not covered by the metaphor's originating paradigm (Ravasio and Tscherter 2004). In our example, the desktop metaphor does nothing to reveal the power of automation that a PC delivers to its user. GUI metaphors are probably best viewed as the expression of the assumptions that software developers hold about the user's interaction with the underlying model – but not, in any case, as a transparent and effective way of allowing the user to engage with the computer's raw power. Metaphors are in this respect paradoxical: what is meant to be a transparent means of interaction with new possibilities of a computational model is in fact an opaque barrier confining the user to a well-rehearsed collection of concepts and processes.

4.1 What Happens at the Intersection?

Paradigmatic regression is not only to be found in graphical user interfaces, we can observe similar dynamics at the level of methodological interaction between or even within research domains. To understand how paradigmatic regression can also occur as a result of the interaction between computer science and textual scholarship, it is useful to view this interaction through the lens of an existing analytical metaphor for such interaction: the trading zone.

The processes at the intersection of research domains (such as textual scholarship and computer science) have been compared to those in trading zones (Galison 2010). Whether they are zones of economic activity or those where methodologies of different fields are amalgamated, pidgins commonly arise in such places. As Peter Galison says: "A reduced common language, which begins with participants in a zone agreeing on shared meanings for certain terms, then progresses to a kind of pidgin and eventually to a creole, which is a new language born out of old ones" (Gorman, Jenkins, and Plowright 2012). Galison also draws attention to the possible existence of visual and mathematical creoles. Indeed, these are not hard to identify in digital humanities: a good example can be seen in the works of Franco Moretti, who has methodologically integrated quantification and visualization methods such as graphs, maps and tree heuristics into comparative literature studies (Moretti 2007). Nor is it very hard to identify current digital humanities as a whole with a new expert community as, according to Galison, they may take shape during the "creole stage" at the intersection of domains. It has been argued that, in creoles of natural language, it is the subordinate group that provides most of the syntactic structure for the creole, whereas the dominant group provides lexical items and concepts. Though Galison provides some empirical observations, it remains an open question whether the same patterns hold for the emergence of methodological pidgins at the interface of different research domains. What interests me here is whether we can indeed observe the formation of a methodological creole in the emerging vocabulary of digital humanities, and whether hints can be found in that vocabulary of a similar regressive dynamic to that observed on the graphical user interface level. It may be that Matthew Kirschenbaum provides us with some – admittedly still anecdotal – evidence of precisely such a dynamic. In a recent article, Kirschenbaum attempts to trace the origin of the label "digital humanities". He identifies a key moment, reported to him by John Unsworth, which seems to have been the tipping point that would propel this label towards its current status of de facto denominator of what then was and still is a non-homogeneous research domain. Unsworth relates the choice of "digital humanities" to a discussion surrounding the title of the Blackwell 2004 Companion to Digital Humanities: "Ray [Siemens] wanted 'A Companion to Humanities Computing', as that was the term commonly used at that point; the editorial and marketing folks at Blackwell wanted 'Companion to Digitized Humanities'. I suggested 'Companion to Digital Humanities' to shift the emphasis away from simple digitization" (Kirschenbaum 2012). Of course we cannot take this as a pars pro toto for the social shaping of the dynamics for a whole field, but it is suggestive. Ray Siemens by no means stands alone in his preference for "humanities computing". Susan Hockey, for instance, titled her contribution to this very companion "The History of Humanities Computing" (Hockey 2004). Significantly, it is the prominent authorities in the field, veritable Nestors, who consistently speak of "humanities computing" – people like Willard McCarty (2005), John Unsworth (2002), and Dino Buzzetti: "humanities computing – I still prefer this designation to digital humanities" (Buzzetti 2012). According to Siemens, the term was "commonly used at that point", yet the publishers preferred the new term in order to broaden the appeal of the concept by choosing a metaphor that felt less challenging. This was a small but pivotal event in the history of the field, which simultaneously points to the state of digital humanities as a methodological pidgin and to an act of paradigmatic regression. The vocabulary juxtapositions in both terms are constructs of a methodological pidgin. Where "humanities computing" suggests an equal interaction or relation between two fields with a stress on computational activity, the term "digital humanities" (purposefully or not) pushes the balance back toward the domain of humanities and subjugates the computational/digital aspect as a partial property of that field. Or in the words of Willard McCarty (2013): "Note, please, the name 'digital humanities' grammatically subordinates the digital […] 'Humanities computing' takes advantage of the ability in English to make a noun serve as an adjective while staying a noun, and it draws upon the participle/gerund ambiguity. But it seems I've lost this contest!"

4.2 The Trading Zone and Digital Textual Scholarship Practice

Scholarly digital editions and the sites where they are conceived and created, virtual or concrete, are themselves methodological trading zones that materialize at two levels. There is a laboratory-like setting tied in a relatively small context to the practice of preparing and publishing a concrete digital scholarly edition – and possibly also the development of a specific technical infrastructure connected to it. At a more abstract level we find a theoretical discussion that connects to the methodological and epistemological histories of textual scholarship, knowledge representation and digital technology. The critical study of these trading zones along empirical ethnographic lines – another approach often applied in science and technology studies – would have much to tell about the methodological interaction between computer science and textual scholarship. Although such an elaborate study has yet to be undertaken, even fairly anecdotal observations nevertheless yield some intriguing insights.

The Huygens Institute for the History of the Netherlands is home to an example of a smaller-scale trading zone in a laboratory setting.[2] The institute encompasses a computer science and software development group that is relatively large by the standards of humanities research, numbering around sixteen professionally trained or educated IT developers. Various members of this group have distinct strengths, such as interface design, data modeling, architecture integration, and text analysis. The group works closely with at

[2] Cf. the information on its website https://www.huygens.knaw.nl/?lang=en

least three researchers who are themselves closely involved in the national and international digital humanities community. Through numerous projects, group members are also in close productive contact with most of the other researchers in the institute and with external researchers active in relevant projects. The projects themselves cover a large part of the spectrum of digital humanities undertakings, from data modeling and repository building,[3] through digital scholarly editions such as the correspondence of Vincent van Gogh,[4] to analytical tool building, of which the text collation engine CollateX[5] is an example (Haentjens Dekker et al 2014).

The research staff of the institute originally had no particular focus on digital or computational activities. In 2005 the institute took the strategic decision to move into the domain of digital scholarly publications as well. The initiative began with the addition of a literary researcher and two developers to the institute. Staff at a related institute, later dissolved, had been developing a "collaboratory" for the curation and analysis of humanities and social science data, which today would be called a Virtual Research Environment (VRE). At the Huygens Institute the part of this environment relevant to the humanities, consisting mainly of a transcription and publication environment for historical texts, was adopted and strongly pushed forward while the social science aspect was eventually abandoned. This eventually became the eLaborate online environment, "in which scholars can upload scans, transcribe and annotate text and publish the results as an online text edition".[6] ELaborate is a Web-based environment where textual scholars find support for basic tasks in creating and editing a digital scholarly edition. A project in eLaborate is essentially a container for a series of scanned manuscript or print text pages that can be arranged arbitrarily in a tree structure. Fine-grained authorization allows one to arrange access or restrictions down to page level and thus to arrange for private, collaborative or fully open edition workflows. A text editor is facilitated to aid in creating diplomatic and critical transcriptions which can be layered with annotations to serve the researcher's or reader's needs. All data is stored and retrievable as XML. In addition eLaborate facil-

[3] https://github.com/HuygensING/timbucto
[4] http://www.vangoghletters.org
[5] http://collatex.net/
[6] https://www.elaborate.huygens.knaw.nl/

itates the automated publishing of Web-based editions and provides a generalized graphical interface based on "fluid" columns. Vertical areas of the screen can be arbitrarily arranged for visualizing the reading text, connected annotations, browsing in the text structure, full text search, and so forth. Given some basic training, eLaborate provides an out-of-the-box solution allowing textual scholars with only average computer skills to create basic digital scholarly editions without much need for technical support.

It is relevant to note that the IT team adopted an "agile" software development methodology. This type of software development takes a manifest user-centered and evolutionary approach to software manufacturing. Short one- or two-week iterations deliver functioning parts of software that are evaluated by the client/user. This ensures the balancing of the software production with the evolving vision and knowledge of the client (Martin 2002). Arguably this methodology feeds into the social shaping aspects of introducing new technologies and methodologies.

A case study of the methodological dynamics surrounding the development of eLaborate serves to show that the trading zone metaphor is not unproblematic. Do the dynamics and interactions in the context – the work site – where eLaborate was developed point to the emergence of a methodological pidgin? Most certainly the developers and the researcher who headed the project started exchanging terminology. The developers began to refer to concepts such as "page", "annotation", "transcription". The researchers grew accustomed to using words such as "user", "interface", "architecture", as well as the vocabulary that is rather typical for the agile methodology used by the developers: "planning game", "iteration", etc. Whether this constitutes a beginning of a methodological pidgin is debatable. The interactions that led to the exchange of vocabulary could equally be attributed to standard development practice in which there is a particular relationship between client and service provider and in which, certainly within agile methodology, the provider normally tries to understand the client's work process and concepts in order to model them into software. The objective of the developers in that case is simply to mimic as closely as possible the concepts the client is using. Arguably this could cause a medium shift in which the researcher ends up with a digital environment that is virtually identical to his or her known analogue work process and material. Once the work is done, the client and developer can go their separate ways, without having essentially influenced the methodologies on either side.

A clearer indicator of methodological change may be the actual loss of lexical items. During the eLaborate project it transpired that an index – in the sense of the keyword reference list in the back of a book – is not a very useful instrument to mimic in a digital environment if the texts at hand are automatically indexed and the interface includes a full-text search function that presents its result as a list of keywords in context. In various edition projects where eLaborate was deployed some friction and dissonance could be observed among users (either textual scholars or trained volunteers who transcribed manuscript material) about the lack of an index, but gradually the use of full text search as a replacement for the index became accepted, even appreciated, once the possibilities for wildcard and fuzzy search were understood. This is notwithstanding the fact that a full text search is not the epistemological equivalent of an index. Current full text indexing technology does not, for instance, facilitate named entity resolution in the same way as traditional indices may. Nevertheless, within projects based on eLaborate the concept of "index" is no longer used except for references to the past; the concept of "zoekfunctie" (search function) seems to have all but replaced it. An index covers of course more important scholarly functions than just finding topics on pages. It also serves as a small ontology to relate synonyms through entries pointing the user to other entries (usually labeled with e.g. "see" or "see also"). Replacing the index with a full text search that lacks such technology would imply a sincere reduction of scholarly function (cf. Rawson and Muñoz 2016). This ontology function that serves as a manual means for coreference resolution was replaced with a categorization function for annotations. Thus, although no scholarly functions were lost the lexical items describing them, such as "index", did disappear. As did the traditional presentation in the case of the index. For textual scholarship and scholarly editing I would argue that the loss of the analogue representation of an index and even the lexical reference to it does indeed constitute a methodological change, as both technique and interface changed. The salient point here being not loss of scholarly function – which did not happen – but loss of terminology as an indicator of methodological change.

The same event serves to show the dynamics of social shaping and regression in a different way. With the indexing technology used in eLaborate – first Lucene[7] and later Solr[8] – it is possible to generate search result lists with text context ranked by "relevance". Although the keyword-in-context search results list eventually found unanimous adoption, the concept of "relevance" became a topic of recurring and fractious debate. Lucene applies a combination of Boolean and vector space models to determine the relevance of documents to a user's query. The Boolean measure selects the documents that correspond to the terms the user wishes to find or ignore. A vector space model is then applied to that selection to rank the relevance of each document to the query. Formally this model determines relevance by applying a cosine measure to the vectorized document vocabulary and query (Widdows 2004). The vocabulary of any text can be expressed as a mathematical vector and the basic trigonometric function of the cosine can be applied to determine the size of the angle between two such vectors. This essentially means that the smaller the angle, the more the vocabularies of two texts are similar. In Lucene this measure is used to determine if requested search terms appear more often in a particular document than on average in the vocabulary of all documents retrieved with a specific query. The more such terms appear in a document, the higher the relevance ranking of that document. It transpired that the textual scholars and other users confronted with this technology were for the most part unimpressed with the relevance ranking, which appeared incomprehensible and alien to them. And although the feature was initially presented in the interface, most edition projects within eLaborate preferred canonical orderings such as sorting by folio number, name of author or text, shelf mark etc. As a result, word-weighted ranking is no longer offered in the editing and publication interfaces of eLaborate, and the researcher in charge of the development confirmed that in the several rounds of open testing that the software underwent, none of the trained users requested the function (Van Dalen-Oskam 2014).

The virtual disappearance of automatic ranking by relevance as a function in the current version of eLaborate is a case of social shaping of technology, and indeed of paradigmatic regression. Ranking by relevance could arguably be methodologically useful for textual scholars who must peruse a large corpus

[7] https://lucene.apache.org/
[8] https://lucene.apache.org/solr/

for occurrences of themes, words and motifs. Even if it is not the default, one would expect the option to be available. Technically there are no barriers to providing the function, as it is the default behavior of the search engine used. In fact, it took additional development effort – though admittedly not much – to provide canonical ordering. Despite all this, the functionality that is standard from the technical point of view is no longer available – a strong signal that the IT developers and the textual scholars found a barrier to knowledge exchange that they were unable to overcome. In other words, they could not create the required methodological pidgin to communicate or appreciate the possible utility of that function.

What is interesting here is not so much the disappearance of relevance-based ranking. There may be valid scholarly reasons to reject such an ordering principle – albeit that these have not been put forward by the users in this case. Rather, it serves as an example in which the pidgin, the "reduced common language" used during the interaction between developers and researchers, was not sufficient to communicate the methodological potential of a relatively straightforward, seemingly useful and non-intrusive method, and so prevented its theoretical consideration. This example shows how difficult it actually is, both for researchers and for developers, to use the trading zone for methodological gain or innovation. The textual scholars involved first needed to know of the existence of such a thing as "ranking by relevance" to be able to recognize its possible methodological potential. Next, to gauge that potential would require them ultimately to drill down to the mathematics of cosine measure for vector comparison and understand how vectors can represent documents. As it has been argued elsewhere in a similar vein, without such a detailed level of knowledge, it is difficult to assess the methodological usefulness of new technologies (Sculley and Pasanek 2004).

It should be noted additionally that this is a small example involving relatively standard digital technology. The syntactical and lexical distance that must be bridged in the case of a project such as Circulation of Knowledge and Learned Practices in the 17th-century Dutch Republic[9] is significantly larger, as in that project correspondences are visualized through network analysis (Van den Heuvel 2012). A sensible understanding of what may be inferred

[9] See http://ckcc.huygens.knaw.nl/

from network visualizations and what this adds in terms of methodology requires a fairly deep grasp of the mathematical models underpinning not only network modeling and analysis in general, but also the topic modeling used to generate the network data (Wittek and Ravenek 2011).

All in all, this raises the question of how much methodological interaction is actualized in a methodological trading zone in the smaller concrete context of DH projects. Some superficial vocabulary is certainly exchanged, of which some may be instrumental in future co-operation for both researchers and developers. However, there is little in the way of deep methodological trading going on. Textual scholars are not providing knowledge about theoretical notions on scholarly editing and literary criticism to developers; and, vice versa, developers are not lecturing researchers about mathematical or computational principles. The common language does no more than create an interface that answers to the perceived needs of researchers in the humanities. The interface becomes an expression of these researchers' conceptions of how the digital technology might serve their purpose.

The methodological gain in this is rather superficial: access and discovery increase in scope, but concepts and processes hardly change. There is a digital translation, but little methodological innovation. The potential or realized methodological innovation furthermore happens rather covertly. In the case of the relevance ordering in eLaborate the potential is there, but hidden – again(!) – by a graphical interface, and by an apparently suboptimal methodological exchange between researchers and developers. In the case of the Circulation of Knowledge project, the mechanics, technology and methodology are almost completely covertly integrated into the resulting digital environment by the computer scientists. A further consequence was that the main technical developer struggled with negative feelings about lack of recognition for methodological merit. The covertness of this methodological innovation is far from trivial. If, as Peter Shillingsburg has pointed out, editions are scholarly and critical arguments about what a textual record means or about how it should be read, then a digital edition is also such an argument (Shillingsburg 2013). Because both interface and model are constituents of the digital edition, they are both part of that intellectual argument. The model – i.e. the combination of the data model and the computer language logic that puts it into action – is entirely conceived by computer science experts. The interface and the view it offers on that model, including the functions of the model it exposes to or hides from the outside world, is to a very large degree conceived by developers and designers. The methodology used for this is effectively inaccessible to the textual scholars, who lack the skills to interpret and comprehend the technologies used. Given that the computer scientists create so much of the intellectual argument pertaining to a particular digital scholarly edition, it would seem that having a sufficiently broad common methodological language is pivotal to digital textual scholarship. But as we can see, our current dynamics of interaction are not helping to create it.

4.3 Trading Theory in the Larger Textual Scholarly Context

Although the trading zone between computer science or digital technology and textual scholarship seems so problematic at the smaller more concrete level, there seems to be no shortage of methodological trading on the theoretical level. Exhaustively detailing and disentangling the intricately intertwined histories of textual scholarship, knowledge representation, literary criticism, computing and digital technologies, is hardly feasible in the span of this chapter. Moreover, creating history often suggests a falsely deterministic account of cause and effect. Nevertheless, it is important to identify a number of key developments. The beginnings of the Internet and the World Wide Web are usually identified with Vannevar Bush's vision of the Memex, an imaginary system to store, track, index and retrieve any information, and – crucially – to rewrite that information and keep versioning records so as to trace the development of our thoughts (Bush 1945). Visions of such knowledge systems reach far further back, however, at the very least to the work of Paul Otlet in the early twentieth century, as has been repeatedly shown (Rayward 1994; Buckland 1997; Vanhoutte 2009; Van den Heuvel and Rayward 2011). It was Theodor Nelson who coined the term "hypertext" and constructed a theory for it, inter alia referring back to Bush (Nelson 1993[1981]). Nelson's attempts at implementing his visions failed to result in successful tools, however. Instead it was Tim Berners-Lee whose team devised – with reference to the work of Nelson (Berners-Lee 1989) – the Hypertext Transfer Protocol (HTTP), which successfully kick-started the World Wide Web. Although sympathetic to his endeavour, Nelson deeply hates Lee's technical solution: "It is vital to point out that Tim's view of hypertext (only one-way links, invisible and not allowed to overlap) is entirely different from mine (visible, unbreaking n-way links by any parties, all content legally reweavable by anyone into new documents with paths back to the originals, and transclusions as well as links – as in Vannevar Bush's original vision)" (Nelson 2010).

Imperfect or not, HTTP technology happens to align nicely with many ideas on the nature of knowledge and text that are emerging in literary criticism, textual theory and semiotics, which increasingly problematize a linear view of text and result in more post-structuralist approaches. George Landow summarizes the convergence: "Hypertext, an information technology consisting of individual blocks of text, or lexias, and the electronic links that join them, has much in common with recent literary and critical theory. For example, like much recent work by poststructuralists, such as Roland Barthes and Jacques Derrida, hypertext reconceives conventional, long-held assumptions about authors and readers and the texts they write and read. Electronic linking, which provides one of the defining features of hypertext, also embodies Julia Kristeva's notions of intertextuality, Mikhail Bakhtin's emphasis upon multivocality, Michel Foucault's conceptions of networks of power, and Gilles Deleuze and Felix Guattari's ideas of rhizomatic, "nomad thought". The very idea of hypertextuality seems to have taken form at approximately the same time that poststructuralism developed, but their points of convergence have a closer relation than that of mere contingency, for both grow out of dissatisfaction with the related phenomena of the printed book and hierarchical thought" (Landow 1994).

Digital textual scholarship and more particularly the digital scholarly edition obviously rely on the technologies delivered by the development of the Internet and the hypertext protocol. In turn, these technologies are rooted in theory which sees the nature of knowledge, information and documents as highly interconnected and referential, or intertwingled and transclusional, as Nelson would in all likelihood phrase it. Peter Robinson expresses similar views when he discusses the idea of "distributed editions", with attribution also to Peter Shillingsburg and Paul Eggert (Robinson 2003). Robinson is interested in the volatile aspects of editions. He posits that readers may become writers too, and proposes that editions may exist in a distributed fashion in an interactive Web-based space. Each reader may have a different representation: "a manuscript transcription from one site, a layer of commentary from one scholar, textual notes and emendations from another, all on different servers around the globe. In a sentence: these will be fluid, co-operative and distributed editions, the work of many, the property of all" (Robinson 2003). According to George P. Landow, this vision is strongly associated with the Docuverse, the ideas on non-linear writing and hypertext systems described by Nelson: "Perhaps the single most important development in the world of hyper-media has been the steady development of read-write systems – of the kind of systems, in other words, that the pioneering theorists Vannevar Bush and Theodor H. Nelson envisioned. Blogs, wikis [...] all represent attempts to bring to the Web the features found in hypertext software of the 1980s that made readers into authors" (Landow 2006:xiv [1997]).

But ideas on more interactive and volatile editions also refer to another complex of theory surrounding the fundamental instability of text. This complex encompasses a post-structuralist view of text where text is not a book but a hypertext, and where hypertext stresses the volatility of text, its heterogeneous, mutable, interactive and open-ended character – ideas rather opposed to that of text as an immutable form enclosed and bound by a front and back cover in a book. This theoretical complex also borrows from ideas on the fluidity of text as expressed for example by John Bryant who calls attention to the perpetual flux texts show trough preprint revisions, revised editions, and adaptations that shape literary works into forms specific to different audiences (Bryant 2002). Similarly, the importance for scholarly editing of the volatile aspects of text is expressed through what has become known as critique génétique, an approach to editing that focuses on the avant-texte, the process of writing and revision that precedes the publication of a book (Van Hulle 2004; Fiormonte and Pusceddu 2006).

The instability and process aspects of text are also important to textual scholarship and the practice of scholarly editing from the point of view of the use of editions: of what happens after publication. The ideas behind hypertext, together with those about read-write systems, also inform ideas concerning the social aspects of text and scholarly editing. Read-write systems facilitate crowdsourcing and thus open up the process of scholarly editing to a potentially far larger source of labour by "expert amateurs" (Hayles 2012) than the individual scholar could provide for (Brumfield 2013). Crowdsourcing engages an audience of users in the scholarly process literally in the avant-texte phase of the creation of a scholarly edition. This potential need not be confined to, say, the transcription stage of a scholarly project. Meanwhile, ideas have been developed on the so-called social edition, which allows readers/users to add their knowledge to the edition and render its creation and use a community event under the guidance of scholarly experts (Siemens et al. 2012). Lastly, the process aspect of text is also highlighted through new computational engagements that readers/users may make with texts and scholarly editions. This aspect was already expressed as early as 1949 through what is now usually seen as the first application of humanities computing: the work of Roberto Busa, which led to the computational means necessary to derive automatically a concordance to the works of St Thomas Aquinas (Hockey 2004). This was the beginning of a long development that prefigured current computer-supported analytic engagement with literary texts such as distant reading, algorithmic reading and big data analysis (Buzzetti 2009; Moretti 2007; Ramsay 2011c; Jockers 2013).

4.4 The Shape of the Digital Edition According to Reality

In short, the interaction between digital technology and textual scholarship places the focus of methodology on both the unstable and fluid aspects of text, and on the process aspects of texts. That is the fundamental tenet that computer science brings to textual scholarship. "Hypertext, unlike print, is fundamentally process- and context-oriented. Following a basic tenet of artificial intelligence theory, it views representing and acquiring knowledge as a problem of defining and searching information spaces, and it recognizes that these spaces and search methods will vary according to the purposes and abilities of particular users" (Edwards 1994). Digital scholarly editions are indeed information spaces. But they are not often information spaces that line up with the theoretical pidgin discussed above. The theoretical notions of textual scholarship, and the scholarly digital edition that we find in the trading zones between textual scholarship and computer science, call for an expression of text and editions through which the information contained in the edition is expressed primarily according to the principles of hypertext. Current reality, however, is very different. In textual scholarship, Internet nodes are mostly placeholders that point via a URL to a digital document or to a digital edition as a whole, as a data silo. The edition of the Van Gogh letters, for instance, sits at the node identified by http://www.vangoghletters.org/vg/ as a fully integrated and monolithic pile of edited text from letters; the pile includes comments, annotations, translations and so on. The finest granularity presented to the network of the web is at the level of the individual letter (e.g. http://vangoghletters.org/vg/letters/let043/letter.html). Even that URL identifies a compound object, that is, a meaningful set of multiple scholarly objects: two facsimiles, a transcribed text, annotations, bound together by an interface that (again following Shillingsburg) represents an editorial argument about what constitutes the digital scholarly edition of this particular letter. According to this argument, there is no need to address the transcription, the facsimile, a particular annotation, in isolation. Most of the digital scholarly editions on the Web are expressed similarly. It is hardly better than a network of nodes in which each node represents a particular edition that is offered as a PDF. This situation renders it impossible to address texts (and thus editions) beyond their graphical interface in ways compatible with a hypertext model.

Digital editions often trumpet the ability to represent text exhaustively, celebrating the fact that there is no need to make decisions on what to leave out (Price 2008). Indeed, it is an asset that digital scholarly editions may be capacious almost without limit. In the case of an important and large tradition of a particular work, this potential may allow for the presentation of all witnesses as items in an inventory, or as a digital archive. Arguably this is not just an asset because of exhaustiveness of representation, but foremost because it allows for the expression of the relations between the witnesses, and thus inter alia the genesis and fluidity of texts – in fact the more process-like aspects of texts – for which the hypertext model as described offers technological expressive potential. In the reality of current digital scholarly digital editions, however, this potential seems seldom realized. A graphical interface will usually allow the user to select and view single witnesses, or perhaps to compare the texts of multiple witnesses, especially if the editor has integrated a collation or comparison tool such as Juxta.[10] The inventory will probably also allow a list of witnesses to be shown in chronological order. The order of that list will in all likelihood be based on a metadata property "date" or similar in the relational database underlying the digital edition archive. The list itself is a generated GUI visualization expressing that metadata. The point here is that a list so represented is not a hypertext representation of the chronological "linkedness" of the witnesses, it is a mere list of individuated metadata. This is different from the idea of hypertext that all information is expressed as machine negotiable nodes and links, so that an expressive network of knowledge is created. This means that the chronological order of the witnesses in this case can only be inferred through human cognition from the metadata based list – it is not represented as knowledge in a computationally tractable form intrinsic to the hypertext medium. Much effort may thus be invested in gathering exhaustive representations of individual witnesses, but if the result of that effort only allows user-level navigation of relational metadata represented as a graphical interface, then the digital scholarly edition is not an effective hypertext knowledge space. Such an edition may still be valuable for the sheer wealth of information, but it remains firmly at the level of document representation for human consumption without integrating the relations between witnesses in a computationally networked representation.

4.5 Regression and Reaffirmation

There is nothing deterministic about technology, and indeed nothing much deterministic about hypertext. As a technology to express a text and to present it in the form of a digital scholarly edition, hypertext has been shaped by the scholarly community into little more than a filing cabinet for self-contained documents. Most digital scholarly editions on the Internet express the particular idea the scholar responsible for the edition

[10] http://www.juxtasoftware.org/

has about what a digital edition is or should be; normally, that idea is a re-representation of the book. We find collections of page-based facsimiles and transcriptions presented as self-contained units, wrapped up in and bound by the front matter that is the interface. There is attention for fluid aspects, and for context. The Hyperstack edition of Saint Patrick's *Confessio*,[11] for instance, explicitly offers its users the possibility to venture from the "centrality of the text [...] through the dense net of textual layers and background information in answer to questions that are likely to arise in their minds" (Fischer 2011). The dense net in question is effectively a star network radiating out from the main page into leaves containing pages of metadata, facsimiles of manuscript folia, or transcriptions of entire texts. Despite the impressive density of information, the information itself is not that densely networked. The relations between the texts and the contextualizing information is described, but not expressed through the "hyper fabric" of e.g. HTTP links. Even so, the *Confessio* is rather an exception to the rule – very few of today's digital editions seem to be particularly concerned with the core ideal of hypertext as an expression of linked information, of process and context.

Most digital scholarly editions, in fact, are all but literal translations of a book into a non-book-oriented medium. Peter Robinson, writing about the distinctions of text- as-work and text-as-document, argues that in the early days of digital editions – roughly until 2005 – scholars would privilege the text-as-work perspective, focusing on the potential of digital technology to express and support the properties of text that construct its meaning (Robinson 2013a:56). In recent years, he continues, this trend has been exactly reversed. More recent digital scholarly editions harness the digital medium rather to represent the text-as-document – the faithful re-representation of a text according to its expression in the physical documents that carry it. As an example Robinson points to the online edition of Jane Austen's fiction manuscripts.[12] Elena Pierazzo, who was deeply involved with the methodological design of this edition, unsurprisingly offers a rationale for a text-as-document approach to the digital edition (Pierazzo 2011). Robinson also notes that many collaborative transcription systems are designed to record

[11] http://www.confessio.ie/
[12] http://www.janeausten.ac.uk

text-as-document: not one of twenty-one tools listed in a survey by Ben Brumfield offers the possibility of recording text-as-work (Brumfield 2013). Indeed it is far easier to point to examples of digital scholarly editions that are in essence metaphors of the book, or in other words: translations of a print text to the digital medium, apparently for no other reason than to fulfill the same role as the print text.

Textual scholarly theory, as has been shown, embraces hypertext as a technology which enables the expression of post-structuralist ideas about information, with a focus on the fluid properties of text. It has often been suggested that the capabilities of digital technologies should become the focus and practice of digital scholarly editing. Despite all this, that ideal is not materializing in the form of concrete digital editions, and for similar reasons to those observed in the smaller context of the eLaborate project. Here, too, we find the dynamics of paradigmatic regression in the professional community surrounding the digital scholarly edition. The methodological potential of information technology is hidden by the incomplete metaphors of a paradigm that is itself reaffirmed by becoming the primary interface to the new technology. Robinson argues that there is a strong continuity of previous contemplation of print editions present in the thinking of those scholars who first conceived of the digital scholarly edition, resulting in a kind of theoretical pidgin that embraces the new technology, but uses it to express digitally a familiar form for the scholarly edition: the printed book (Robinson 2013a:60). The print edition in that digital translation is a metaphor, but one that begins to hide hypertext's native potential for expressing referential and conceptual links between texts. The graphical interfaces of digital scholarly editions almost all refer strongly to this book metaphor, reaffirming thereby the paradigm from which that metaphor springs. In the end, the use of the technology has shaped it into a tool to recreate that which is already well known. It is also worth noting that the de facto lingua franca of current digital scholarly editions, TEI-XML,[13] is instrumental in this reaffirmation. As an encoding language it is geared fully towards describing text-as-document. Although not graphical in nature, TEI is thus an interface that, like graphical interfaces, hides many of the essential networking and process characteristics of hypertext. Instead, TEI-XML, with its text-inward orientation, print-text

[13] http://www.tei-c.org

paradigm and hierarchical structure focus, continuously reaffirms the view of the digital edition as representing a text-as-document.

4.6 Beyond the Book?

There is nothing deterministic about the Internet. The paradigmatic regression we currently see in the digital textual scholarship community is a clear demonstration of that. The textual scholarship community has devised a methodological pidgin that exploits a new technology to express a well-rehearsed paradigm of scholarly editing. Yet this must not be where the methodological shaping and disciplinary trading stops. The theoretical concepts pertaining to the fluidity of text are clearly important to the textual scholarly community, but they still need to be brought fully into the concrete methodological pidgin that is currently geared towards representing a text-as-document, rather than toward text-as-process. As long as scholarly editors keep producing digital metaphors of the book, this will hardly happen. Both textual theorists and computer science practitioners must intensify the methodological discourse to clarify what existing technology is needed to implement a form of hypertext that truly represents textual fluidity and text relations in a scholarly viable and computational tractable manner – a hypertext language inspired both by computer science and textual scholarship. Without that dialogue we relegate the raison d'être for the digital scholarly edition to that of a mere medium shift, we limit its expressiveness to that of print text, and we fail to explore the computational potential for digital text representation, analysis and interaction.

Chapter 5

Code, Scholarship, and Criticism: When Is Code Scholarship and When Is It Not?[1]

> "l'historien de demain sera programmeur ou il ne sera plus"
> (Le Roy Ladurie 1968)

5.1 The Softwarization of Scholarship

There is no single easy definition of code. Code can be regarded as a new semiotics with its own literacy (Knuth 1984; Vee 2013). It can also be seen as a mode of existence of software, which at least has two such modes: a textual and a processual dimension (Hiller 2015). The textual dimension is connected to code in its form of source code, which is the text produced by a programmer in a formal language that – once interpreted by a computer – results in executable software. The processual dimension is connected to the execution of code as a computer program, which invokes also a performative nature. Mark Marino has argued that we should "analyse and explicate code

[1] A previous version of this chapter has been published as Van Zundert, Joris J., and Ronald Haentjens Dekker. 2017. "Code, Scholarship, and Criticism: When Is Coding Scholarship and When Is It Not?" *Digital Scholarship in the Humanities* 32 (Suppl_1): i121–i123. https://doi.org/10.1093/llc/fqx006. This article was co-authored, but the greater majority of research and all of the writing was done by the first author, supported by comments from the co-author. Some sentences were redacted for clarification and a paragraph more precisely delineating "code criticism" in the context of this book was added.

as a text like any other, 'a sign system with its own rhetoric' and cultural embeddedness" (Marino 2006). For the argument presented here code is regarded as source code mostly. That is, code in its guise as blueprint for a program that can be executed. With regard to such code the question is when a particular piece of code acquires a scholarly nature. What properties or qualities force us to consider the source code of software as a scholarly object of study? And if we can determine those properties, then how do we evaluate the scholarly merit of these code objects? As we shall see in answering these questions, the operative aspect of code ("what it does") turns out to be of essence too.

However, before turning to such issues a pivotal question needs to be answered first: why does code deserve scholarly attention at all, for in past decades it has not been a given that code is indeed of scholarly interest (cf. for instance Bauer 2011). A rationale for the humanities to consider code as a scholarly object of study and to consider code as a scholarly object itself can be argued along two approaches at least. The first is related to a general "softwarization" of society as described by, inter alia, Berry (2014). The second is a more specific realization of this trend that relates to how we understand tools as instruments applied in research.

The "softwarization" of society that Berry argues has also been vividly described by Jones (2014) who refers to it as "eversion". This "eversion" is a term coined in the 2007 novel *Spook Country* by William Gibson, who is also famed as the author of the cyberpunk cult novel *Neuromancer*. The concept of "eversion" serves to identify the process of cyberspace turning itself inside out and flowing out into society beyond the point where either is truly separable (Jones 2014:28). Where prior to 2007, cyberspace was an alternate but separate and virtual reality into which human existence in some visions might eventually even transmigrate; after 2007, the ubiquity of access points to the digital realm, the omnipresence of embedded computer technology, and the primacy of digital streams as carriers of information let the worlds of the virtual and of reality merge and intersect to a point that it is very hard to tell them apart. Jones marks the appearance of the smartphone around 2007 as the point of articulation between these realizations of digitality. Berry describes in a similar vein the pervasiveness of computation and digital information, and questions it from a perspective of critical theory. At this point in time, cultural artifacts and the processes of creation and interpretation tied to these artifacts are as much digital as they are not. Arguably therefore, the humanities should concern themselves with the humanistic status and interpretation of such artifacts and with the creative processes that they result from.

Concerns with how pervasive forms of computation affect society are raised often in the context or as a result of critical theory. People such as Coyne (1995), Berry (2014), Marino (2006), and McPherson (2012) approach the digital from a socio-philosophic vantage point and interrogate how social context shapes software and how it in turn affects society and the relation of humans to digital technology – mostly with the aim to critically examine whether the technology liberates or limits the potential for personal, cultural, or social freedom and development. The omnipresent and massive impact of digital objects and processes on society and culture should also be of concern to the humanities in and of itself because it deeply affects the socio-technical processes by which cultural artifacts are created and interpreted, thus affecting the object of study of the humanities.

There is also a more narrow methodological rationale for the study of code in the humanities. Just as software and digital information pervades society, it emerges in the humanities virtually everywhere. It appears both as source and object of study, e.g. in the form of digital data and information, and as resource, in the form of tools and infrastructure. If code is thus an emerging object *and* method of study – such as text is for the humanities – it should arguably be the subject of scholarly examination. A rejoinder to this is the often-invoked metaphor that one does not need to understand an engine to drive a car. That however is an improper metaphor for software. An article by Ian Hacking (1981) “Do We See Through a Microscope?” will be useful in understanding why this metaphor is erroneous, even pernicious. Hacking’s argument centers on the question of how to establish the reality of what we see with a microscope. Fundamentally there is no way of knowing this. As humans we cannot empirically verify or testify that there is an object under the microscope when it is too small to sense. We trust however that the theory of optics holds, and that therefore the image we perceive is true to the nature of the object. We accept and trust that the way light passes through a system of lenses is accurately described and predicted by the theory of optics.

Yet this remains "just" a theory, despite the fact that it has repeatedly held up under testing. But exactly because no one has yet been able to prove that the theory is incorrect regarding the behavior of light in a microscope, we trust that what we see is what is actually there. Or in Hacking's words: "It may seem that any statement about what is seen with a microscope is theory-loaded; loaded with the theory of optics or other radiation. I disagree. One needs theory to make a microscope. You do not need theory to use one".

Hacking's remark sounds very similar indeed to "One needs computer literacy to make software. You do not need computer literacy to use it". The crucial difference is that code and software are not governed by a law of nature in the same way optics are. If the curvature of a lens is incorrect a user will get a foggy or blurred picture of a plant cell (for instance). But it will remain a blurred picture of a plant cell. No matter how broken the lens, it will not transform a picture of a plant cell into a picture of the facetted eye of an insect. Software code by contrast is "written" or "built" by humans and is not bound to natural rules of proper and verified behavior. Most mobile phones carry an inbuilt lens these days, with a camera "app" to take pictures. It would be rather easy to change the camera's software in such a way that whenever a user takes a picture, some random picture on the Internet would be presented as the photograph. Thus what Hacking justifiably concludes for microscopes on the basis of a general and well-supported theory of optics does not hold for software. In both cases there is a situation of trust. In the case of lenses we trust that a well-verified theory of light and optics will hold and that the nature of light and its interactions with materials will not change overnight. In the case of software there is a trust that the result of creative coding work will do what the creator of that work says it will do. But software tools are lenses of a different kind: at the time of writing according to TextMate (a robust no-nonsense text editor for Mac OS) this text up to here has 1,167 words, according to MS Word it has 1,174. If something as deceptively simple as counting the number of words in documents gives different results in different pieces of software, how do we trust complicated topic modeling software like Mallet that produces hundreds of clusters of terms as suggested topics found in a corpus? Software is governed not by laws of nature, but by the rules that are programmed into it by the engineer, that can be set by anyone having access to the design process of the software, and that can result in incredibly complex heuristics and algorithms. This fact should by itself warrant some systematic approach to critiquing code. But especially now that more digital tools are getting integrated into the methodology of humanities, the adequacy and validity of analyses depend to a certain extent on an adequate understanding of such specific rules.

5.2 Scholarly Assumptions in Code

To make this more concrete let us study the case of CollateX (Haentjens Dekker et al. 2015). CollateX is software under active development at the Huygens Institute for the History of the Netherlands.[2] CollateX is – as the name suggests – a collation engine. The core of CollateX consists of an algorithm – that is, a defined sequence of precisely specified steps that produce an output (Schmidt 2016). Algorithms as mathematical and programming concepts have a long history of themselves (cf. Bullynck 2016). Also some inroads toward the study of algorithms have been made from the humanities and social sciences, most noticeably from the perspective whether and what knowledge of algorithms is pertinent to humanities (Seaver 2013; Schmidt 2016). Here I am not interested so much in a mathematical proof of CollateX's working, but in a similar vein as Seaver we want to know how particular assumptions of the developers about text and text scholarship become inscribed in the algorithm that makes up the core of CollateX. In the following all statements on the CollateX software pertain to the 2.0.0 version of the Python port available on the Python library repository PyPI (Python Package Index).[3]. The open-source code of CollateX is available under GPLv3 license in GitHub.[4]

CollateX's algorithm, if given a number of texts that are largely but not exactly the same, will align the parts of texts that run parallel, or "match" as this is usually called. For instance, if the algorithm is given the following texts:

```
1. the black cat hops over the red dog
```

[2] http://collatex.net/
[3] https://pypi.python.org/pypi/collatex
[4] https://github.com/interedition/collatex/tree/master/collatex-pythonport

```
2. the white cat hops over the dog
3. the black cat hops over the red cat
```

It would align these "witnesses" (as variant texts are usually called in textual scholarship) as follows:

```
1. the | black | cat hops over the | red | dog
2. the | white | cat hops over the |  –  | dog
3. the | black | cat hops over the | red | cat
```

Collation is a scholarly task central to the field of textual scholarship, which is itself concerned with establishing a solidly argued representation of a given text. Because the process of collation is labor-intensive, repetitive, tedious, and error-prone (Robinson 1989), it is a good candidate for automation. As with all software, any such automation will result in an implementation of an algorithm that to a certain extent rests on particular assumptions (Lehman and Ramil 2000). The algorithm of CollateX makes three tacit assumptions on the heuristics of alignment:

1. It is desirable to minimize the number of differences between witnesses
2. Phenomena that are shared across most witnesses should be preserved
3. The number and order of witnesses are arbitrary

Furthermore the algorithm of CollateX is based on at least one axiom that states that it is computationally infeasible to distinguish between a transposition and a combination of substitution and deletion. That is, if the algorithm finds the following alignment:

```
1. the cat hops over the black dog
2. the dog hops over the black cat
```

It is nigh impossible for any computational algorithm to decide whether the "cat" and "dog" in the first sentence were switched (textual scholars speak of a "transposition") or if either of them was individually replaced (i.e. substituted by a consecutive deletion and addition).

The issue here is not whether these assumptions are correct, but rather that they exist in the code as such. They represent rules and choices that could have been different as a result of different scholarly reasoning and argument. Assumptions are inscribed tacitly in code rather than being explicitly mentioned or described by it. It would be very hard indeed, even for skilled engineers, to reverse engineer or read the code so that these assumptions become apparent. Yet they are part of the very rationale behind the mechanism that fulfills the scholarly task of alignment.

In the case of CollateX, the aforementioned assumptions may not be shared by each textual scholar. They are indeed not laws of nature, nor are they generic mathematically proven principles. Especially the axiom concerning transpositions could be subject to scholarly debate. A human reader will apprehend quickly that in the example above, the "cat" and the "dog" were transposed. But unless evidence external to the texts is presented, fundamentally this is not deducible with complete certainty – it could have been that the cat was replaced with another dog. The apprehension of the human reader is in fact an assumption, conjecture based on intuition and experience. A rule of thumb could be that when more words are involved in a potential transposition (so longer fragments are switched) and the fewer words there are between the two potentially transposed fragments, the likelier it is that a deliberate transposition occurred. It is unlikely that an author would for instance switch around a "the" at the beginning of a text with a "the" at the end of that text. If we find "It was a dark and stormy night" in one witness at the beginning of a text, and in another witness at the end, it is more likely that deliberate transposition was the cause. It would be very time-consuming to take this rule of thumb into account when computing the alignment of witnesses because the number of comparisons that need to be performed by the code would grow exponentially. Hence the axiom: it is fundamentally impossible to know from the texts alone if a transposition happened, and it is computationally highly costly to compute all potential transpositions; thus, it is computationally infeasible to distinguish between a transposition or two independent substitutions.

The third assumption, which posits that the alignment should be independent of the number and order of witnesses, is also debatable from the perspective of textual scholarship. Suppose that it is clear from external evidence – e.g. from the bindings of a manuscript or the type of materials used – that a particular witness is older than any other. In those circumstances it becomes a legitimate scholarly question whether that witness should be a guiding text, or a "base text" as it is called when specifically used as a guide for decision-making in the process of alignment (Roelli 2015). In unmarked situations, however, it is assumed that baseless collation is preferable (cf. Andrews and Macé 2013). During the development of CollateX, great care was taken therefore to prevent it from presenting a result that is in some ways biased or colored by the particulars of one specific witness. Indeed this feature became a "unique selling point".

The contention based on the above is that code through its mathematical and algorithmic origins does not acquire some inherent objective and neutral correctness. Instead the construction of code is situated and depends on the assumptions of its builders, be they subjective, supposedly objectified, or scholarly. In this respect code and software cannot escape what has been similarly found for data and facts. There is no such thing as "raw data" (Gitelman 2013), rather data and facts are carefully constructed (Bowker 2006; Latour and Woolgar 1986). This does not deny the potential solidity of facts, but it calls attention to the situatedness of their creation. Even what "constitutes" data is dependent on context and often even far from clear, especially in the humanities (Borgman 2015; Kouw, Van den Heuvel, and Scharnhorst 2013). Within the digital humanities this has given rise to criticism on how data should be understood, on data representation (Drucker 2011), and on the use of (standards for) digital formats (Vitali 2016).

5.3 Scholarly Code Criticism

The assumptions that underpin the code of specific software in textual scholarship ought not to be the idiosyncratic musings and intuitions of individual programmers. In the case of CollateX assumptions were inferred from close and repeated conversations between the lead developer and a variety of textual scholars who had a particular interest and experience with text collation. These assumptions are in this sense a result of aggregated, carefully interpreted scholarly knowledge re-inscribed in code. It is this process of aggregation, interpretation, and re-inscription of knowledge that lends the code of CollateX a particular scholarly nature. Insofar as interfaces and code bases can also be thought of as arguments (cf. Galey and Ruecker 2010), it is these assumptions by which the code of CollateX captures and adds to the ongoing scholarly debate on collation. As argued above however, the argument that code makes is very implicit. How can scholars – or for that matter other programmers – examine and critique this code and these assumptions as an integral part of academic discourse?

This question points to a clear need for a method or a framework within the humanities to systematically explore and validate scientific software engineered for and used in the humanities. No such agreed upon formal method or framework for critical evaluation of code exists. Nor is there an agreed upon method to share any results of the critical evaluation of code. As Mark Marino has stated in a field report on critical code studies (CCS): "there remains a considerable amount of work to develop the frameworks for discussing code" (Marino 2014). Marino's report presents a concise history of CCS that suggests that they are indeed an application of critical theory. CCS studies the social context and processes surrounding code and its creation. A good example is McPherson's 2012 contribution to *Debates in Digital Humanities*, titled "Why Are the Digital Humanities So White? or Thinking the Histories of Race and Computation" (McPherson 2012). Read superficially it is an article that will make many (white male) computer engineers roll their eyes and sigh: sure, UNIX is racist. However that is not McPherson's argument: "I am not arguing that the programmers creating UNIX at Bell Labs and in Berkeley were consciously encoding new modes of racism and racial understanding into digital systems. [...] Rather, I am highlighting the ways in which the organization of information and capital in the 1960s powerfully responds – across many registers – to the struggles for racial justice and democracy that so categorized the United States at the time. [...] The emergence of covert racism and its rhetoric of color blindness are not so much intentional as systemic. Computation is a primary delivery method of these new systems, and it seems at best naive to imagine that cultural and computational operating systems don't mutually infect one another."

Another clear concern of CCS is the aesthetics of code and code-as-text. Marino (2006) is interested in reading code as text: "I would like to propose that we no longer speak of the code as a text in metaphorical terms, but that we begin to analyze and explicate code as a text, as a sign system with its own rhetoric, as verbal communication that possesses significance in excess of its functional utility." Given this proposition it is understandable that CCS is fascinated with poststructuralism-inspired uses and interpretations of code, such as Alan Sondheim's concept of codework that mixes computer code and text, and in which computer code thus additionally becomes a medium for artistic expression (Wark 2001).

Although critical theory inspired code criticism arguably should be part of any framework for evaluating the scholarly qualities of code in the humanities, the approaches and examples from the field of CCS also still leave a lot to be desired. To understand this, consider a remark Evan Buswell made during a HASTAC 2011 CCS event (Marino 2014). Buswell stated that CCS cannot only deal with the arbitrary elements of code because that would relegate code criticism to aesthetics only. This was a reaction to Mark Marino's suggestion to try to read code as text and to use code variables as meaning forming elements to see how this would give expression to the meaning of code. Buswell was quick to note that variable names are arbitrary because of an ubiquitous code mechanism called indirection. Variable names are wrappers and boxes: what is printed on them needs not to have an intrinsic relation with what is in them. Thus if one reads in e.g. JavaScript:

```
var welcome_message = 'Welcome to my homepage!';
```

It simply means that there is a variable with the name "welcome_message" that holds the text "Welcome to my homepage!". However, that name is arbitrary. The code:

```
var bananas = 'Welcome to my homepage!';
```

creates the same result (which is that there is a variable with the text value "Welcome to my homepage!"). Thus the name of the variable does not entail anything about the value of the variable or its meaning within the code, or beyond.

Mark Marino's argument was based on the assumption that developers usually use "speaking names" for variables, precisely because it keeps the code somewhat readable, and hopefully clear to other developers. Under these conditions variable names may indeed reveal something about the assumptions and norms connected to the context in which the code was developed. If the variable was named "opening_sentence" instead of "welcome_message", this may reveal something about the intention or frame of mind of the developer. The former might indicate an engineer foremost focused on text structure, the latter might suggest that the programmer was thinking more about user interaction.

Thus there is certainly reason to do as Marino suggests and to read code also simply as "a text". However, code is a text that performs. It also represents a program that can be executed, and fundamentally variable names do not reveal this performativity. They do not reveal necessarily the aim of the code, nor how it operates. Thus, as Buswell concluded, student engineers may learn from CCS to carefully choose their variable names because they will be working with culturally sensitive programmers in various cultural contexts and settings – but "all the while there will be an invisible line between CCS and CS, protecting the core from the periphery, insulating and separating from critique the power structure of code itself, and constructing a discourse of good code and bad code to go along with the discourse of good business and bad business that tends to dominate naive anti-capitalist critique".

Before I claim that a solid framework for criticism of source code applied in scholarship is lacking, as I will, I should explain what this means exactly. Of course there is a tremendous amount of work done in both software studies and media studies to construct frameworks to critique software. Lev Manovich's *Software Takes Command* (2013), *Expressive Processing* by Noah Wardrip Fruin (2009), and the already mention work by Tara McPherson (2012) testify to this achievement. But to call these works "code criticism" would be somewhat of a misnomer. And in fact that is not where these scholars position themselves: they talk about software and media criticism. These works study the effects of software, which is the performance aspect of code: that what the user sees, the interface, its uses and affordances. They study the performative aspect of software as a performance: how software creates effects in people's behavior, work, opinions, and so forth. They also study software "in the wild": the socio-political effects of mainstream software used in industry, as games, and as a tool of personal and institutional productivity. But what they do not do, is critique or evaluate the actual source code of software, the particular kind of text that formulates the behavior of software. Also their object of study is that of software in society, not specifically that of source code applied as a scientific tool in the humanities (or sciences). Hence they are not studies of scholarly code and they are not code peer review, but they are studies of social effects of software at large.

This is what I mean when I speak of "code criticism" in particular: a scientific framework for peer review of source code that is written specifically for and applied in textual scholarship (or other scientific fields) to evaluate that source code for its scientific reliability, implied methodological choices, and implicit scholarly interpretations. My concern is the fact that we increasingly use bespoke code – i.e. tailor made, one off applications to serve specific purposes and specific concerns in particular scholarly editing or research. For the critique of this kind of code, in textual scholarship and by implication the humanities more in general, we lack an established theoretical and practical framework.

As a framework for code criticism CCS seem to lack a rigorous method for examining and critically interrogating actual code beyond reading the "code as text". In addressing this it would make sense to draw a parallel between the interdependent relationship of textual criticism and literary criticism on the one hand and between code criticism and CCS on the other hand. Literary criticism is the application of critical theory and aesthetics to literature. It is occupied with the interpretation of literature, its contextualized meaning, its cultural inwardly and outwardly influences, its development over time, etc. Textual criticism is less about reception, meaning, cultural situatedness, and writerly text.[5] Rather it is the critical skill of establishing a well-argued representation of a text. Though "fact" in the light of post-structuralist theory is a problematic term to say the least, it is not unreasonable to posit that textual criticism is pre-occupied with scientific textual fact finding and accountability: textual criticism tries to establish as close a "factual" representation as

[5] For a concise explanation of "writerly text" see Mambrol (2016). This aspect is also dealt with in some more depth in the next chapter.

possible of a text through a scientifically accountable process (cf. McGann 2013).

Textual criticism faces its own particular challenges resulting from digitality. Since authors turn to personal computers and text processing software for text production, textual criticism – used to an almost exclusively physical materiality of manuscript and print publications – is confronted with the realities of digital materiality. Scholars in this field are therefore augmenting and adapting existing methodologies to this new reality. This includes, for instance, new approaches to the preservation of personal archives left by authors on hard drives (Grigar et al. 2009; Kirschenbaum et al. 2009). It also includes adaptation of scholarly methodology aimed at studying the genesis of authorial documents, since genetic stages change from "manuscript draft" and "print proof" to revisions stored in for instance .docx files, the standard file format for more recent versions of MS Word (Ries 2010; Buschenhenke 2016).

Arguably a framework for scholarly evaluation of code could encompass components of CCS and components that are more directly aimed at factual code review – similar to how text critique encompasses literary criticism and text criticism. The CCS component would focus on answering questions of broader socio-technical impact. For instance, is there an ideology underlying this code? What are the cultural assumptions and biases apparent in the code? What was the social context of its development? The code criticism component would aim at critically examining the actual code and its scholarly or scientific intentions. What is the stated purpose of this code? Which scholarly task – perhaps in relation to the concept of scholarly primitives (Unsworth 2000) – is it trying to accomplish? How well is it accomplishing that task? What concepts and relations are modeled into the code?

Code criticism in this sense would first of all be pragmatic. If literary criticism asks the question "What does this mean?" and CCS ask "How does this code affect us?", then textual criticism asks "What was written here?" and code criticism asks "What does this code do?" Code criticism could deliberately pose deceptively simple questions to code because this aids in revealing the scholarly nature of code. As an example one can compare CollateX with eLaborate, another tool developed for use by textual scholars.

ELaborate is a tool for digital transcription created and actively maintained by the Huygens Institute for the History of the Netherlands.[6] Transcription is undeniably a scientifically valid and valuable primitive of humanities, especially with regard to scholarly editing and philology. Is therefore eLaborate to be deemed a scholarly tool? The software supports the scholarly task of transcription. Does this mean that the software and the code itself are scholarly and thus examples of scholarship? The key is in the distinction between enabling and performing tasks. ELaborate enables the scholarly task of transcription, but the transcription itself and all the scholarly skills and decisions tied to it are still performed by the user. ELaborate is not somehow magically more adequate in registering the keystrokes of a scholarly editor than WordPress, MS Word, TextMate, or any other text editor. It has a number of features that greatly facilitate the task, and allow the editor to really focus on it. Otherwise it does its best to get as much out of the way of the scholarly editor as it can. It has less feature clutter than for instance Word, it has a centralized and institutionally backed repository for all its data, it is Web-based, and so forth. In comparison with other tools this means that there is seemingly always one specific feature that makes eLaborate a better fit for the scholarly task than most other text editors. Yet it would be hard to argue that the code propelling eLaborate is scholarly in itself and by itself.

This distinction of the scholarly nature of code is based on the question whether "scholarly decisions and choices are delegated to the code level". This implies that there is no absolute certain measure that can tell whether code is scholarship because establishing the scholarly nature of the code depends on convincingly arguing that such decisions indeed were delegated to the level of code, which is an argument that always should take into account the situatedness of both the code's development and its intended purpose. In the case of CollateX this can indeed be argued considering the currently pervasive practice of aligning variant texts by hand in textual scholarship. Decisions currently normally taken by a scholar are delegated to the code, as the algorithm results in a possible alignment of variant texts

[6]http://elaborate.huygens.knaw.nl/

that implies decisions on which words between variants match. In contrast, the code of eLaborate does not effect or propose as a result such choices or decisions that currently would be deemed scholarly significant in textual scholarship.

In all, this then does not preclude at all that the "code building" connected to the development of eLaborate can be a scholarly valuable act or contribution. The development of eLaborate is certainly a valuable scholarly achievement: scholarly thought and argument were part of the process of its creation and the design of its specific functionalities (Beaulieu, Van Dalen-Oskam, and Van Zundert 2012), and much subsequent scholarship was enabled through the use of eLaborate. Because scholarly argument at some level is involved, it is still relevant to critically examine eLaborate's interface, features, and capabilities. This would be tool criticism however, not code criticism. Obviously tool criticism might at some point very well be integrated into the approach suggested here, but that is beyond the scope of this argument. In the case of CollateX decisions that are understood as scholarly responsibilities in the current context of textual scholarship are delegated more extensively to the code itself than in the case of eLaborate. Therefore, unlike eLaborate, the code of CollateX "performs" a scholarly task: based on the tacit assumptions built into its code, the algorithm of CollateX independently makes scholarly informed decisions – or rather proposes these, as the decisions until now are always ultimately corroborated by a scholar. It is arguable therefore that the code of CollateX in the current context of textual scholarship is endowed more with a scholarly nature than the code for eLaborate. That in fact the code of CollateX represents scholarship and is itself a scholarly object. This is no different from a monograph or print edition, each one a scholarly object whose scholarly nature arises from the arguments they constitute and represent.

Critically examining this argument and the scholarly nature of the code itself is not straightforward however. Above the mostly tacit nature of scholarly assumptions built into code was already pointed out. But code is unintentionally covert in other ways as well. Engineers often talk about the "model" that underlies their code. "Model" is rather a "hopelessly polysemous" word, as Willard McCarty remarks (2005:27). There is extensive literature meanwhile also in the field of digital humanities on the meaning, purpose, applications, and epistemology of models and modeling pertaining to scholarship, data, and code, notably McCarty (2005), Flanders (2012), Jannidis and Flanders (2013), Ciula and Marras (2016) – but there are many more. Obviously there is a relation between the analytic model and data used by researchers and the data and object models constructed through code by programmers. The collection of digital object (definitions) and their relations expressed in code make up the domain model, as defined by Fowler (2002:116), which essentially expresses the programmer's "understanding" of the models used by the researcher(s).

In the context of source code creation by software engineers, the model component of the code is thus that which comes to represent the conceptual or phenomenological model of the problem domain. That is, the concepts, the relations, and the operations that mimic the problems, objects, and processes the software developers are trying to automate or solve on behalf of a client or, in our case, a researcher. In the case of eLaborate, the model has coded objects such as "Transcription" and "Annotation". Annotation objects in the code may have associated functions or methods, such as "create", "update", or "delete". Of course all the components are needed in a meticulously orchestrated combination to make the software function; all components are in that sense essential to it. Not any framework for code criticism can therefore conveniently eschew some part of a body of code. However, the components that capture the domain model are probably the most closely associated with inscribing the conceptual model of the researcher into code, as opposed to data storage components or visualization components. To complicate matters even more maybe, visualization obviously constitutes a transformation of the data that by itself is modeled too. Visualization transformations therefore also constitute an interpretation and argument about data. Like tool criticism however, the problem of interface critique is out of scope here – even though it is readily imaginable that the "code" that drives visualizations could be subject to code criticism within a framework of code criticism framework.

Even if we boil code criticism down to critiquing the domain model, effective code criticism may turn out to be a strenuous activity. Domain models may be hard to gauge or peruse from the code that is eventually published, and they can be unintentionally obfuscated. Models may be as tacitly expressed in the code as the assumptions underpinning it, or they may be confusingly cloaked by different code expressions. Part of the algorithm of CollateX, for instance, is based on a decision tree. This tree is used to recall which decisions were made by the algorithm to come to an alignment between witnesses. If a new witness needs to be added into the comparison, previous alignment solutions can be compared to favor one solution. For reasons of performance (i.e. speed) and scalability, the decision tree is not expressed in the code as a tree, however. Instead the engineer chose to use a matrix that will deliver the same power of decision but at a very much lower performance penalty. Reading directly from the code, it would be hard, or at least considerably confusing, to see that a matrix was used to perform the function of a decision tree.

Thus just as with the variable names that can be arbitrarily chosen and thus obfuscating, code may be for good reasons unintentionally enigmatic. The nature of code in this sense seems to resemble poetry more than prose. Poetry sometimes intentionally uses enigmatic or hermetic language, forcing the reader to reread and rethink possible meanings. Code will in general be less intentionally enigmatic, but will sometimes be no less hermetic. Sometimes such hermetic code becomes a goal in itself, for instance when coders try to come up with "oneliners": tiny algorithms of one line of code that perform certain, sometimes incredibly complex tasks. Arguably one of the best-known examples of this "onelining" is "10 PRINT CHR$(205.5 þ RND(1));: GOTO 10", to which even a full book publication was dedicated (Montfort et al. 2012). Such witty solutions may earn particular admiration of other coders, the solution being regarded as a particular "elegant" one. Yet the "coolness" of the solution may result in code that is particularly obfuscated and hard to read. Also the actual algorithm may be counter-intuitive even if mathematically highly efficient. This is arguably the case with the Quicksort algorithm, which is an algorithm for sorting that carves the series to be sorted into subseries.[7] This "divide and conquer" strategy results in a performance gain (i.e. sorting speed) several magnitudes larger than the approach generally found to be more "intuitive", called insertion sort, which involves starting at the top of the list and inserting each item in the series in its "nat-

[7] https://en.wikipedia.org/wiki/Quicksort

ural order" position.[8]

5.4 Criticism in a Continuum of Literacies

How then do we critically examine code that may be particularly hard to read, scrutinize, and understand? At the very least, an attempt should be made at reading the code, even if simply to establish the degree of readability of the code, because this is valuable information for criticism too. If the code is nigh incomprehensible, what does this mean? Can the reasons for possible intentional obfuscation be deduced and/or reasonably established? Is the illegibility a result of unskilled coding? Obviously inline comments and external documentation should offer help in determining the intent of the code as well. Also establishing the software development methodology used can reveal useful insights. There are various methodologies to build software, from highly formalized and rigorous to fully pragmatic "cowboy coding". Some methodologies are bound to be a better fit than others for the heterogeneous nature of humanities data and research questions (Van Zundert 2012).

Mostly however: why not talk to the creators of the code themselves? Assuming that engineers indeed apply current so-called "good practices", software development is a highly dialectic practice. The adequacy and effectiveness of code are mostly determined by how well the model that is inscribed in the code fits the domain model of the problem or task that the software was developed for. To deduce a best-fit model, engineers should go to great lengths. Analysis and design for modeling in most current software development methodologies will involve deep client and/or user interaction. That is, during the design phase, engineers will interview the client over and over again to explore the exact properties of the domain model. And during any implementation phase, engineers will in all likelihood repeatedly expose the execution of the code to the scrutiny of the researcher and will adapt the design iteratively to what the researcher reports back as to shortcomings, omissions, etc. Thus the model is designed, tweaked, and tuned in a continu-

[8]Cf. https://en.wikipedia.org/wiki/Insertion_sort

ous communicative and dialectic feedback cycle between developers and researchers.

If the engineering of a model is governed by dialectic, the most adequate mode of scholarly code criticism could be parallel. Code as an argument can be adequate but obscure, and in such cases a good way of establishing the model tacitly underlying the code could be to reverse engineer it through discourse. Thus by reversing the dynamic of the dialogue, we may understand software in the same way as its development was articulated and argued: by a deep and continuous, even "intimate" as Frabetti (2012) suggests, dialectic. What is reversed during the phases of creation and criticism is the role of the interviewer and interviewee.

A similar parallelism and mirroring arises in another potential avenue for critically examining code. It is a good practice in code engineering to develop not just code but also a test suite for that code.[9] A test suite or harness is a set of tests expressed as code that can be run to check that software is working correctly. Engineers can in this way guarantee the correct working of the code. Tests are used to check the workflow, to test against critical conditions, to inspect certain expected output for given input, to test the formal constraints of a model, and so forth. It may turn out to be as valuable for code criticism to examine the test suites that accompany code as the contents of the code itself. Much may be gauged from these tests about assumptions, corner cases, conditions, flow, limitations, and intent of the code.

But an even more intriguing application of test suites might be for scholarly code critics to develop these suites themselves. Currently test suites and automated tests for software are tools of the engineer. But there is no reason why the frameworks that help engineers to control, check, and validate their work would not be used to probe, explore, and test the same software by code critics. Instead of facing the engineer, test harnesses might just as well face the critic and user. Several people involved with CCS have expressed similar ideas. Montfort et al. (2012:322) speak of studying "software by coding new software". David Berry refers to such possible test suites as "coping tests" (Berry 2014).

[9]https://en.wikipedia.org/wiki/Best_coding_practices

The possible application of code to test code, to create test suites to examine codebases as a form of humanities informed criticism, can also be cast as a continuum of two literacies. Three decades ago Donald Knuth believed that the time was "ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature" (Knuth 1984). His language called WEB lets the same program produce working code as well as an explanatory narrative about that code. WEB however, did not find a broad audience, neither in computer science nor in the humanities – with the odd exception (e.g. Huitfeldt and Sperberg-McQueen 2008). Knuth was interested in code as a form of literature and in writing software as a specific kind of literacy. In other words, he was interested in how two kinds of literacy, that of computer language and that of human-authored text, could merge. As a proponent of computational methods in the humanities, Franco Moretti appears to agree that this merging has a lot of promise (Dinsmann 2016). Literacy enables one to write *and* read, to express *and* inquire. From these perspectives the understanding of the literacy of code and the literacy of text as different and opposed literacies seems artificial and intellectually lazy. Instead, to develop a valid and adequate mode for scholarly criticism of code, they need to be understood as variations within a continuum of literacies (cf. Kittler 1993b).

These notions so far are theoretical. Also in the realm of this chapter, I have only been able to sketch the outlines of a possible practical approach to code criticism for code created in the humanities. Based on these more theoretical ideas currently work is undertaken to explore how text and computational literacy may amalgamate into a concrete method for code criticism. This work takes the form of an iPython Notebook in development[10] that reports on the investigation of the code underpinning a publication by Ted Underwood and Jordan Sellers (Underwood and Sellers 2016). A follow-up publication will be dedicated to this explorative practical code criticism case study.

[10]https://github.com/interedition/paceofchange

5.5 Conclusion

Code criticism and code peer review are hardly even nascent in the humanities and digital humanities. Some work has been done in the realm of CCS, but these fledgling approaches have focused primarily outward from code and have considered code mostly as a culturally situated part of a larger socio-technical system. Almost no examples of thorough code criticism exist that regard code from a humanities methodological point of view. The type of criticism that asks: what is methodologically expressed here, how is it argued, and how can we validate it? Given the large ramifications that digital information and software have for humanities sources, resources, and methodology, this situation is rather surprising, and methodologically unhealthy. In this chapter I sketched the outlines of an approach that would do justice to the work that has been done in the realm of code criticism but that would also self-reflectively turn criticism toward the code that promises new tools to the humanities.

For centuries, argument, logic, interpretation, and reason have been both the means to put forward results in the humanities as well as the tools to judge those results. Humanities methodology is highly self-reflective. Methodology now increasingly means digital methodology, but that does not imply that critical self-reflectivity should disappear: there is no self-evident correctness of technology just because it is digital technology. Much work still needs to be done to remediate the critical aspects of humanities scholarship into the digital realm. This is a critical task digital humanists should be aware of.

Chapter 6

Author, Editor, Engineer: Code & the Rewriting of Authorship in Scholarly Editing[1]

> "Programmers: stop calling yourselves engineers."
>
> (Bogost 2015)

Like the previous chapter this chapter examines the relation of software creation to textual scholarship, and more in particular to the creation of (digital) scholarly editions. But now I take the concept of authorship as my specific analytic vantage point. Authorship speaks to both the creation of software and to the creation of scholarly output in the form of text editions.

My resulting claim is this: writing code is a form of authorship; but in the majority of cases coding in a scholarly context is a form of unclaimed or, worse, misappropriated authorship. The authorship of code inserts at least one layer of interpretation into the process of scholarly editing in comparison to traditional modes of editing. This creates two problems. First, the text and its interpretation get potentially even more fluid and unstable, which is contrary to the aim of philology and scholarly editing, whose aims can generally be understood as an attempt to stabilize the text. This notion of stabilizing

[1] A prior version of this chapter was first published as Van Zundert, Joris J. 2016. "Author, Editor, Engineer – Code & the Rewriting of Authorship in Scholarly Editing." *Interdisciplinary Science Reviews* 40 (4): 349–375. http://dx.doi.org/10.1080/03080188.2016.1165453.

the text is highly problematic in itself, as we shall see. Adding the authoring and performance of digital code into the process of editing is likely to further destabilize the text and its interpretation. Whether this is desirable or not is irrelevant to my argument. What I argue – and this is the second problem – is that this process goes unchecked by our current scholarly processes, because it is largely not recognized either by scholar or programmer as authoring or as scholarly editing. Its potential influence on scholarly editing, irrespective of its aim, therefore goes largely unevaluated.

6.1 What Is an Author?

Any determination of the relation of authorship to either scholarly editing or programming requires an understanding of the history of the notion of authorship. Therefore the first part of this chapter occupies itself with a short historical overview. This overview is necessarily limited, as it is quite possible to fill books by themselves on the matter. I also limit any related claims to "authorship in the West", which might be roughly understood as authorship in Europe before the Spanish conquest of the New World, and in Europe and North America after. Even that may be overstretching such claims because many cultural, religious, and sociopolitical aspects are related to attitudes towards authorship, and these contexts differed of course through times in these regions as well. But still some generalization is allowable. In any case I would carefully want to stay away from claiming anything related to e.g. attitudes concerning authorship in other regions, such as Africa or Asia, because I have not perused any sources on such matters.

Over the course of history the perception of authorship has been very tightly related to fluctuating ideas on who exactly produces textual meaning and interpretation. As a consequence the appropriation of authorship has been closely associated with an authority over epistemological claims, ontological claims, and with claims to truth. In contrast to classical, mediaeval, premodern, and even later times, today authorship seems to warrant far less authority over the interpretation of a text and the claims to truth it might make. In science and scholarship that authority arguably has shifted foremost to evidence and peer review. It has been eroded to a large extent by critic and reader in the case of literature. This undermining of authority is however a relatively recent matter – and as we will see a contentious one.

As an historical overview much of the following borrows heavily from Burke (1995) who has described the history of authorship in the West as a continuous tension between the presence and absence of the author. Throughout this history views on authorship have swayed back and forth from author essentialism to almost complete impersonalization. Plato can be said to have sided with the idea of impersonalization. To Plato the personality and poetic imagination of an author was entirely unimportant. The world of forms was just a shadow of the ideal world of ideas, so any intervention by creative imagination on the part of the author when describing the world would just result in mere shadows of shadows. True knowledge could only be attained by disinterested rational enquiry. By advancing the idea of catharsis Aristotle on the other hand defends the empathetic nature of literature and poetry and thereby the empathetic role of the author. These perceptions of the role of the author are connected with epistemological claims. That is, essentially they are asking whether authorship is a means to knowledge – to which Plato answers "no" as it can merely record, and even that it can do only inadequately; but to which Aristotle answers "yes", the empathetic imagination attached to the act of authoring may bear on knowledge. As such the roles of authorship and of the author become attached to epistemological claims, to claims about truth. With the rise of Christianity the debate about these roles becomes even more pivotal, as authorship pertains to the Scriptures and to hermeneutics, which is the theory of interpretation of these scriptures. How have the scriptures been conceived? Were the authors mere vessels to be filled with Divine inspiration, non-interfering bodies that merely moved the quill while sacred words passed through? And if this went for the scriptures, what of written interpretations such as those of Augustine? To explain the epistemological power of hermeneutics, the idea that divine truth can be revealed in an author is essential. This *auctoritas* becomes the keystone of the early mediaeval epistemology: an author that can be named and referred to as the authority for given knowledge. Yet, as the Middle Ages draw to a close a theory of hermeneutics emerges that holds that truth and knowledge may be acquired also through careful reading and reasoning. No longer are epistemology and truth given, instead they are now derived. Both epistemological modes – that of given and derived – rely strongly on the presence of an author and the assertion of related biography. It is necessary to refer to an auctoritas to claim a truth with any authority. And in case a statement is not based on external authority, to be able to make claims on truth and truthfulness the moral and literary status of the author must be unquestionable.

Thus the philosophical legacy of the middle ages results in two basic tenets for authorship. One is reasoning, the other is divine inspiration. Ratio will eventually become the basic tenet of Enlightenment. As the religious undertone retreats, inspiration eventually becomes equated with Romantic creative imagination. Romantic authorship still involves overtones of divinity. But no longer is this (exclusively) a religious divinity. The divinity and sanctity have become aspects of an innermost personal genius that moves the author. Burke in this respect refers to Edward Young who speaks of a "stranger within" and a kind of "inner God". At the end of the nineteenth century the author is strongly immanent in both types of authorship. Rationality culminates inter alia in positivistic naturalism. Inspired authorship is found from romanticism, with its most intimate expression of the most intimate experience, to expressionism.

For all these forms of authorship, it is pivotally important who speaks, who authors, because the text lays a claim to a truth. Claims of philosophical truths, of truths in natural philosophy and the Enlightenment, theological claims, the social and political claims of Positivism, the claim that Romanticism makes to an inner truth of the author as a person.

The notion of possible subjectivity of interpretation only comes into the story with Albrecht Husserl's phenomenology and Martin Heidegger's ontological hermeneutics. These new theories of perception and interpretation erode the authority of the author. Late nineteenth century literary history and criticism were mostly concerned with establishing criteria of aesthetics – often motivated by a need for nationalistic literary canonization. Authorial poetics and the intent of the author were important aspects in this type of criticism. But the acknowledgement of the subjectivity of interpretation causes a "de-centering" of the author. This is reflected strongly in the New Criticism that accepts biographical information about an author as contextual evidence for a possible interpretation of a text, but generally assumes that the true authorial intent cannot be established. In the words of Wimsatt and Beardsley in "The Intentional Fallacy": "the author's intentions in writing are neither recoverable nor pertinent to the judgment of the work" (Wimsatt and Beardsley 1954[1946]). From "The Intentional Fallacy" it is only a small conceptual step to Roland Barthes' "The Death of the Author" (Barthes 1967). Here Barthes argues that interpreting a text solely along the lines of its author's biography, is in fact limiting that text and the possible interpretations of it: "We know now that a text is not a line of words releasing a single 'theological' meaning (the 'message' of the Author-God) but a multi-dimensional space in which a variety of writings, none of them original, blend and clash. The text is a tissue of quotations drawn from the innumerable centres of culture." In *S/Z* he would expand this idea into the notion of the "writerly text", by which he means that any text is primarily a site for intertextual meaning production at the hand of the reader and critic (Barthes 1975).

An equally oft-cited text to motivate the move that decentered the author is Michel Foucault's "Qu'est-ce qu'un auteur?" (Foucault 1983[1969]). In this essay Foucault abstracts away from the author as a particular historicized person and instead talks of "author-function". This signifies that it is not the person of the author that is of essence, but the identification of what authorship as a transcendental phenomenon tries to accomplish. Foucault then argues that the most salient feature of authorship is its capability to initiate new discourse and new discursive practices. Thus Freud's authorship was not so much significant because it defined a set of texts authored by Freud, but because it allowed a new form of discourse to be developed that could be referenced to by "Freudian psychoanalysis". To quote: "[...] within the realm of discourse a person can be the author of much more than a book – of a theory, for instance, of a tradition or a discipline within which new books and authors can proliferate". This feature of the "author-function" is domain and medium-independent according to Foucault. It also extends towards the domain of the technological: "I am aware that until now I have kept my subject within unjustifiable limits; I should also have spoken of the 'author-function' in painting, music, technical fields, and so forth".

Jeremy Hawthorn has argued that both Barthes and Foucault explored in these ways the possibility to read literary works with an attitude similar to that adopted by readers of non-literary and scientific work (Hawthorn 2008:74). For Barthes the author is a founder of language, a "Logothete" (Barthes 1989:3–5). Through authorship and the creation of an oeuvre the author develops a new semiotics that only shares its surface with a known linguistics of human language. But just as numbers do not need the intentionality of a factual consciousness, Barthes argues, the meaning of this language is not bound to the faculty of the author. The language can be interpreted well without it. Similarly, by defining the "author-function" as the founder of discourses, Foucault essentially contends that the meaning and relevance of a text is established by the agents active in the discourse it pertains to, and not solely or even at all by the author: re-examining the discourse founded by Marx modifies Marxism (Hawthorn 2008:75).

"The Death of the Author" and "What is an Author?" have often been seen as a final death certificate for the authoritative role of the author. Though obviously any text requires a writer, many post-structuralists have argued that the biography of the body writing a text is in no sense pertinent to the interpretation of the text. According to Compagnon (2004:45) the American literary critic Stanley Fish exemplifies this "dogmatic relativism" and "cognitive atheism" as he: "maintains, in radical opposition to the objectivist pleading for an inherent and permanent meaning of the text, that a text has as many meanings as readers, and that there is no way to establish the validity (or invalidity) of an interpretation. From this point on, the reader is substituted for the author as the criterion of interpretation". Compagnon and Burke find the reasoning underpinning this absolutism rather reductive and confused though. "What is at work in this slippage is an egregious simplification of the immense problematics of the relations between self, ego, transcendental ego, consciousness, knowledge, and creativity" (Burke 1989:122). In the topos of the death of the author, the author in the biographical or sociological sense is confused with the sense of the author's place in the historical canon, and the author's intention in the hermeneutic sense, or intentionality, as a criterion of interpretation: Foucault's "author function" perfectly symbolizes this reduction.

Burke and Compagnon both call on cognitive and linguistic philosophy to argue that you cannot go around intent. That very faculty is applied to constrain the possible meanings of a text. No matter what subsequent context dependent interpretations add to the polysemia of the text, in part the meaning that such an interpretation is derived from is shaped by the intent of the author. This is different of course than saying that there is a one to one relation between this intent, the text, and its interpretation. The author may have been unsuccessful in the expression, the reader may have been unsuccessful in decoding the expression, or indeed she may have "rewritten" the text reading it. The point of the matter is: intent is a necessary prerequisite for meaningful text.

Post-structuralist debate with its insistent emphasis on the decentering of the author could suggest that authorship also has lost its authority over the claims to any truths a text makes. However, just as intent cannot be so easily circumvented, neither can this authority be fully circumvented. Much weakened perhaps by the reader's ability to interpret, but an authorial authority still exists; not just any claim or any truth can be read into a text with as much validity. A certain authority thus remains with the author. These claims may be hard to gauge in literature, as we may need to dig through different layers of meaning within fictional, or textual, worlds (cf. Hawthorn 2008:76) intentionally clouded by unreliable voices (e.g. in Nabokov's *Lolita*). But we assume and expect them to be clear and open in scholarly and scientific work. The most salient – most defining – feature of authorship therefore is its intent to make a claim to some truth.

Thus the report of the death of the author, her intent, and her authority was an exaggeration. On more precise inspection – e.g. the investigations of Burke, Compagnon, and Hawthorn – it seems that the death of the author was actually an epistemological metaphor rather than an ontological truth. The author needed to be decentered, even discarded, for a period of time for the post-structuralist debate to be able to develop the reader as a site of meaning production and interpretation (Burke 1989:57). However, having firmly established that subjective role it was time for the author to step back into the picture and claim back her own subjective role. This process is nothing more or less than an example of what Foucault argued in "What is an Author?" and to which that text itself contributed: that authorship is a socially constructed role (Foucault 1998:213). The role of the author and of authorship is brought in line with the purpose attributed to a text by its audience. Thus the lesser authority of authors on claims of truths in scientific texts and the remaining to a certain extent of such in fiction. It is not either author or reader, but it is both of them negotiating the meaning and the interpretation of a text and the validity thereof from their individual situatedness. In the words of Compagnon: "The text [...] has an original meaning (what it means for a contemporary interpreter), but also later and anachronistic meanings (what it means for subsequent interpreters). It has an original signification (putting its original meaning in relation to contemporary values), but also subsequent significations (always putting its anachronistic meaning in relation to current values)" (Compagnon 2004:61).

This process is much reminiscent of ideas on the social construction of technology that assert a reciprocal non-deterministic relationship between culture and technology (Klein and Kleinman 2002). Very roughly: culture influences technology, technology influences culture. Similarly there exists no determinism in interpretation. The text does not signify the "single 'theological' " intent of the author, to speak with Barthes. Rather the interpretation is shaped by author, and critic, and reader. It is a shaping that is collective yet distributed over time and geography, situated individually as each agent is. A social construction of interpretation. Interpretation to which the intent of the author is pertinent if – certainly in the case of literature – elusory.

6.2 The Author and the Editor

How does authorship relate to scholarly editing? Obviously both the creation of a literary or scientific text and the scholarly editing of the same – not uncommonly with several hundreds of years between those events – involve authorship. But what exactly is the authorship appropriated by the scholarly editor? Editing in general is the preparation of a text such that it may be printed or otherwise published for reception by an audience. Scholarly editing concerns itself with establishing, curating, and studying the record and archive of historical texts – in practice most often the legacy in some form of dead authors. If we follow Greetham (1994) on this the work of a scholarly editor entails at least the following activities: finding a text, reproducing the text (which involves reading, evaluating, and transcribing the text of the original document or documents), criticizing the text, and editing the text. This process then results in a new representation of the text of the originating document(s). Although the phases of the process may be readily defined, each text is unique and each text confronts the textual scholar with complex and puzzling traces of its own genesis. In practice therefore the process of scholarly editing involves a series of choices that affect the eventual expression and representation of the text. Because of the many idiosyncrasies a particular text may hold, it is generally accepted that consistency and accountability should be the hallmark of a proper scholarly edition. Or in other words: the editor should be making the same choice in the same situation and explaining this to the reader – for instance, telling the reader that the editor chooses to represent the glyph “u” consistently as “v” where in a mediaeval manuscript that is used for the consonant.

Usually a scholarly editor will provide not just the conscientiously edited text, but considerable additional paratext as well. An introduction is often added arguing the text’s historical or cultural importance, or a specific peculiarity that motivates it being edited and republished in a scholarly fashion. The editor may put the text presented into context, explaining the genesis of the text, clarifying biographical details on the author, or expounding the text-theoretical underpinnings of the edition. Such additional knowledge may be part of the published text (e.g. the introduction) or it may be presented in additional scholarly articles, blogs, interviews and so forth. As an example we may point to Hans Walter Gabler’s (et al.) edition of James Joyce’s Ulysses (Joyce 1984). Here all variants from the available manuscripts of the text have been painstakingly perused to aggregate an impressive synoptic edition reflecting the text as – according to the scholars – it was intended by the author. The same scholarly edition may serve to point out the predicament any scholarly editor puts himself in when creating a scholarly edition. Gabler’s edition prompted a series of reviews and scholarly articles concerning its controversial editorial methodology (Lernout 2006). This debate has been regarded as a confrontation between Anglo-American and French-German editorial philosophies, but it also signifies a fundamental problem in philology and textual theory which is the assertion – as Dilthey would put it – that a text can at best be understood, but not explained (Compagnon 2004:41). The actual and exact intent of an author of fiction is in the end impossible to retrieve. Even what seems like a straightforward case of harmonizing orthography – such as the distinction of u/v – results in the undecidable pondering of ambiguous expressions (cf. for instance Gerbenzon 1961). The reader who is not a medievalist might not know, but the glyphs 'u' and 'v' have been used interchangeably at times and sometimes it cannot even be decided if one or the other was meant by a copyist of a medieval or classical manuscript, let alone that it would be decisively decidable what the original author intended to write. If even such "trivial" cases are hard, what about the interpretation of the work as a whole? This is where the practices and principles applied in many scholarly editions clash with ideas on the subjective interpretability of text and the impossibility of establishing authorial intent (Wimsatt and Beardsley 1954[1946]). Ultimately we cannot know with any certainty from the text or the biography the intent that is part of an author's transcendental ego. As both Burke and Compagnon argue we run the risk of making anachronistic interpretations of a text because of our distance to the time and context of the author. And we may be fooled because the text is an inadequate expression of the governing intent of the author. All editorial decisions are thus at risk of not (well) representing what was intended by the author. All editing is interpretation, an estimation of what the text was intended to express. Scholarly editing then, lays a claim to a truth, which is that the delivered edition is the most warranted expression of the text by scholarly standards. Scholarly editing is not the same as the authorship it claims to represent, it is however authorship that claims to represent that authorship. In this respect the authorship involved with creating a scholarly edition is very close to Harold Love's definition of revisionary authorship (Love 2002:46), so that in effect the editor becomes the self-asserted proxy of the author. A cynical interpretation would cast the scholarly editor as an imposter exploiting the text of an author. Indeed the image of the "editor [...] as a brutish interloper forcing his interpretation on the defenceless text" led Murphy (2008) to muse: "Much of the most advanced contemporary theorizing about editing has suggested that the editor should follow the author into his grave".

Broadly speaking scholarly editing formulated two responses to this post-structuralist challenge. One response was to fully embrace the ideas of the reader as the site of production of interpretation. In Greetham 1994, David Greetham's rejoinder to the post-structuralist challenge was still to state that although such schools have perhaps effectively described the textual phenomena, they "have not yet produced a critical vehicle for representing them in a scholarly edition" (Greetham 1994). At that very time though, the Internet had solidly established itself as the world's newest, fastest growing, and most intrinsically explicit networked site of knowledge – and George Landow (1994) was publishing on the analogies between post-structuralist theory and the new digital networks: "Discussions and designs of hypertext share with contemporary critical theory an emphasis on the model or paradigm of the network [...] The analogy, model, or paradigm of the network so central to hypertext appears throughout structuralist and post-structuralist theoretical writings." Landow refers to Heinz Pagel to explain the theoretical appeal of the network as structure or paradigm "to those leery of hierarchical or linear models". A network has no top or bottom, but an arbitrary number of connections between nodes that "increase the possible interactions between the components of the network". More importantly: there is no root node, no node to top all others: "[t]here is no central executive authority that oversees the system".

The perceived decentering, anti-authoritarian nature of digital networks and hypertext appealed to a new generation of New Criticism oriented scholarly editors. As Murphy (2008:298) argues the " 'Barthian' argument is a stock phrase in pro-digital editions discourse". Jerome McGann claimed that any editorial act, even any textual engagement, resulted in a new text with its own unique features and conditions (McGann 1991). Ideally the goal of textual scholarship then should be to create an "archive" representing this text pluralism that not just decentered the author, but ultimately even the text. A "fully networked hypermedia archive would be an optimal goal", an "archive of archives" (McGann 1995). Other scholars such as Robinson (2004), Peter Shillingsburg, Van Hulle (2015), and Siemens et al. (2012) also embraced the apparent openness and bias-free nature of the digital environment that would allow for the untainted representation of all textual witnesses. The rhetoric of this response is liberatory, reminiscent even of Critical Theory, with Robinson most comprehensively summarizing its ideology: "All readers may become editors too" (Robinson 2004).

The other response was, rather in contrast, a reinforcement of the primacy of the document. And somewhat surprisingly perhaps this reaction has, inter alia, also found Jerome McGann as a proponent. For in a recent publication McGann (2013) seems to return solidly to the primacy of the archival task of scholarly editing as a form of documentary editing that exerts authority over material fact. He rejects the post-structuralist project altogether and reasons that philosophy is actually a subroutine of philology that is concerned with testing, reconstructing, or falsifying its objects of attention. The primary task of textual scholarship should concern the "archive" of what is known or has been known, which is both concrete in the sense of (digital) "edition" and a metaphysical ideal: "Philology is the fundamental science of human memory". McGann reasons against new media technologies as ephemeral and less adequate for any archival task: "The room of philology is more extensive than the internet room because it is a fully historiated enterprise – because it is, so to say, conscious that its current use depends upon the strength and depth of the belatedness it can never escape. Shaped to a vast presentness – blogging, texting, tweeting, LinkedIn, and Facebook – the internet makes it difficult for us to see and to remember two things about itself: that, as a knowledge tool, it is 'before anything else, memory'; and that, as a memory system, it will keep on forgetting" (McGann 2013:344). For McGann this is a reason to deplore current masters and PhD programs that do not adequately prepare for a philological understanding anymore. Although he recognizes that a focus on Python, XSLT, and GIS is important, "[...] one might better think that descriptive bibliography, scholarly editing, theory of texts, and book history are now even more pressing programmatic needs".

Well-known practitioners and theorists of textual scholarship such as Hans Walter Gabler and Elena Pierazzo likewise retreat forcefully to the primacy of the document. Robinson commented: "For Pierazzo, the possibilities of the digital medium have created new possibilities, which enable the making of detailed digital representations of the document using complex encoding. [...] For both Gabler and Pierazzo, digital editing is rooted in the document. [...] It is difficult to imagine a more articulate and forceful exposition of a theory of digital editing as focused on documents than that given by Gabler." (Robinson 2013a:110–111). Together with McGann these scholars assert that the materiality of the document allows a claim to a truth as philologic fact – and with this simple gesture McGann reduces the impact of hermeneutics and the pertinence of authorial intent for the text to mere afterthought: "For the philologian, materials are preserved because their simple existence testifies that they once had value, though what that was we can never know completely or even, perhaps, at all. If our current interests supply them with certain kinds of value, these are but Derridean supplements added for ourselves" (McGann 2013:346).

Furthering his argument in a following publication McGann contends that this establishing of philological fact should be "less under the rule of theory or idea, and more as a regimen of careful practice" (McGann 2015:215). This practice is the thorough and rigorously methodic study of text. Studying for instance "a passage in a book few people read or perhaps have ever heard of – perhaps a book in a dead language – trying to say something accurate and truthful about it" (McGann 2015:217). Calling upon Milman Parry as a witness McGann concludes that this is the task of the scholar: the commitment to an "impossible truth [...] the truth, the whole truth, and nothing but the truth" and "the obligation to protect human memory from neglect and erasure – as much of it as possible". However, on reading Parry we find that this truth is not some documentary truth. Parry made a very strong epistemological claim in a time when he felt that history and historical texts were abused as propaganda for truths that they would not support. Parry's truth is a deep understanding of the text from a fully historiated contextualization of the text and the authorial intent: "So, gradually, we learn to keep ourselves out of the past, or rather we learn to go into it, becoming not merely a man who lived at another time than our own, but one who lived in a certain nation, or city, or in a certain social class, and in very certain years, and sometimes – when we are concerned with a writer in that whereby he differs from his fellow men – we must not only enter into the place, the time, the class – we must even become the man himself, even more, we must become the man at the very moment at which he writes a certain poem" (Parry 1971:410 [1936]). Parry's claim is historically situated itself of course – it is 1936 – but it is nonetheless a very strong epistemological claim to truth on behalf of the scholar, based on the assertion that it is possible to reconstruct historically situated meaning and authorial intent. Parry is talking very much about interpretation, and not about material philological fact. He is talking exactly about interpretation when he says: "But the scholars must see that they must impose their truths before others impose their fictions" (Parry 1971:413 [1936]).

The 'documentary turn' of inter alia Gabler, Pierazzo, and McGann can be characterized as an attempt to confine text and textual scholarship to some material constraints or – in the case of Pierazzo's rationale for digital documentary editing – a limiting digital model thereof. The purpose of that attempt is to escape the dreaded temporally and conceptually unconstrained realm of situated interpretation – a fleeting interpretation that, moreover, must yield some authority on interpretation to an authorial intent that, following Compagnon and Burke, cannot be established with any certainty. Likewise the "archival turn" represented for instance by Robinson, Shillingburg, and Van Hulle is an attempt to evade the problems of interpretation and authorial intent by embracing the full authority of the reader over these. Robinson (2013a:119) follows Paul Eggert in this, who – after tracing the ideas of Heidegger, Saussure, Foucault, Barthes, and Blanchot – settles on Adorno's idea of negative dialectic. This is the idea, roughly, that one is defined by what the other is not. In this sense, the document is the textual site where the agents of textuality – e.g. author, copyist, editor, typesetter, and reader – meet, and where they bind each other in a dynamic of perpetual redefining each other. And according to Robinson the "digital medium is perfectly adapted to enactment of editions as an ever-continuing negotiation between editors, readers, documents, texts and works" (Robinson 2013a:127).

Thus both responses by textual scholarship to the post-structuralist challenge attempt to evade as much as possible a claim to the interpretation of the text – for the more interpretative an edition is, the more the scholar asserts to have knowledge about authorial intent. Yet, a retreat to philological fact does not resolve the predicament. If the genetics of a text are not difficult at all, the role of the scholarly editor is trivial. But if they are hard – such as in the case of the Joyce edition of Gabler, or in the case of intricate phylogenetics of mediaeval manuscript traditions (cf. e.g. Reenen et al. 2004) – much interpretation is required to establish the text. But strong interpretation comes at the risk of anachronistic and subjective readings. Nor does the "archive", if understood as an (electronic) archive of "philological facts", resolve the issue. For the same holds – in extremis it would be a repository of (digital) image facsimiles, and again the role of the editor would be trivial. Moreover, it is questionable how fair it is to leave such an archive to the reader without any guidance. The skills, experience, and expertise of the scholarly editor would be withheld from the reader. For this reason Murphy argues for strong interpretation: that editors must presume to know what readers ought to know (Murphy 2008:306).

If scholarly editors do not want to theorize themselves out of existence – a risk Murphy points out – they had better re-evaluate their role as interpreter. Why be so shy about interpretation, if it cannot be prevented anyway? Why retreat in the background as much as possible, as a subservient muted voice disclaiming any pertinence to interpretation? Why not rather fully recognize the inevitable and claim full responsibility for revisionary authorship informed by historical expertise and literary skill? Why not, in so doing, liberate and acknowledge both the partial authority of the scholarly editor as well as the partial authority of the reader over interpretation of the "archive"? With Rorty I would argue that actually the biography and intent – indeed ultimately fundamentally unknowable in any objective sense – should not matter too much. Why would we limit the meaning of a text in search of some unattainable objective philological documentary fact? "The contrasting view is to assume that the works of anybody whose mind was complex enough to make his or her books worth reading will not have an 'essence', that those books will admit of a fruitful diversity of interpretations, that the quest for 'an authentic reading' is pointless. One will assume that the author was as mixed-up as the rest of us, and that our job is to pull out, from the tangle we find on the pages, some lines of thought that might turn out to be useful for our own purposes" (Rorty 1988:34).

Supposedly objectified material and philological fact-based scholarly editing has developed into a dogma. To elaborate on Rorty: no doctrine is dangerous, but the thought that textual scholarship depends solely on one doctrine – that the hand of the editor can behave such that it does not taint the meaning of the text – is. Any editing is an 'intrusion' on authorial intent. Even the question whether a deceased author would have wanted the text to remain in existence cannot be satisfactorily answered by philology. The more sensible role for philology is thus interpretation – interpretation that cannot in any case be circumvented. It must therefore always be clear that the text presented is an interpretation. We should not be made to think that we are reading James Joyce's *Ulysses*. The reader ought to know that she is reading *Ulysses according to Gabler*.

The documentary turn in particular, but also the archival turn, are attempts to establish a methodology aimed at stabilizing texts. But text apparently does not want to be stabilized. Each and every editorial act results in a new interpretation and likely in most cases a new text. The textual record is thus ever both expanding and lossy, ever changing – indeed like autopoietic systems as McGann suggests (McGann 2004). The authorship that editorship is, is both preserving the text and paradoxically expanding its fluidity. It is reluctantly revisionary, yet revisionary. Scholarly editors argue their text in a scientific fashion, the hallmark of which is accountability. And accountability is nothing more or less than arguing a claim to a truth. The editor claims some representation of some text to be the best possible given evidence obtained in a scientifically controlled process of perusing sources, collating them, weighing possible meanings and representations, and so forth. Such a claim should be made clear, open, and unambiguous to be scientifically viable. Rather than hiding behind the author's name, the scholarly editor ought to reveal him or herself utmost, because the editor has quite some claims to account for.

At the very least the scholarly editor makes a claim to actually having edited the text. In textual scholarship that is often also a claim to the text or the work itself. Before an edition is published, a textual scholar usually implicitly claims a more or less exclusive right to be working on an edition of a text. Unless commercial copyright is involved, that claim cannot be legally enforced, but legal intent is usually not the motivation for these claims anyway. This tacit claim to be exclusively editing a text also exists with long since out-of-copyright texts, like mediaeval manuscripts for instance. A scholarly editor is "working on" a text; he or she usually intentionally lets this be known through papers and presentations on conferences or a website, actively claiming that text as a personal or institutional site of research. It is generally considered poor academic form or outright impolite to simultaneously create alternative or competing scholarly editions. This is different of course when an edition itself has become a historical text, at which point the object of that edition – that is, the original text – may be considered to be in dire need of a new scholarly edition, which paradoxically introduces all the same problems of authorial intent in a recursive fashion.

The appropriation of a scholarly edition is also a claim to qualification and adequacy as to the work of editing the text. Not just anybody has the necessary skill and qualifications to edit a text properly and in a scholarly manner, as Greetham (1994) and others (e.g. Shillingsburg 2006; Murphy 2008) argue. Connected to this claim of craftsmanship is of course the claim to McGann's impossible textual truth. Explaining or reconstructing the text is impossible, both because production and reception of a text are situated, and because in all probability essential evidence for the authorial intent is lacking and impossible to attain. Yet customarily scholarly editors of historical texts will argue from contextual and biographical evidence the motivations and intent of an author. However deconstructivist one wishes to turn, this historicized situatedness of the author is valid information pertaining to the interpretation of a historical text.

A fifth claim is constituted because the appropriation of an edition is both an honorific and a passport to academic credit. An editor often goes by the name of the text he or she is working on, so that the editor can be known as "the editor of the Decameron" for instance or "the editor of James Joyce". A scholarly editor can establish her name in the field in this manner and often there is a Matthew-effect connected to appropriating a particular prestigious edition. In other words: the appropriation of an edition of a particular text procures significant (academic) value. There is the intrinsic cultural and social value for instance of the curated text. But there is also the economic and social value connected to the publishing of a new edition of a classic text. Connected to the effort of creating an edition are the academic values of scholarly skill, knowledge of textual editing, and the subject knowledge that a scholar builds. Lastly there is of course the academic status that the editor derives from all this.

Foucault argued that the authority of the "biographical" writer (i.e. the author and his intent) on interpretation was less in the case of scientific texts. Interpretation in this genre was mostly the prerogative of the discourse and those involved with that as a whole. The author's name was but a handle to the text (Foucault 1983:12–14). For Foucault the appropriation of the text was less important than the ability to read the language of the literature in a mode that was independent of the author – a mode of reading more akin to the functioning of text in science, and that stresses the discursive aspect of textuality (Hawthorn 2008:74). But appropriation is paramount for the revisionary authorship that scholarly editing is. In general, the appropriation of scholarly and scientific works is rather more important than Foucault would have it, due to the social aspects of science and scholarship involved with establishing truths and building reputations (Latour 1987). This is not because Foucault is wrong about the discursive aspect of scholarship, however. The controversy around Gabler's Ulysses edition shows that the revisionary authorship of scholarly editing leads to new discourse. The scholarly articles and alternative editions that followed Gabler's edition exemplify the social construction of interpretation. Appropriation is important because scholars are pressed into the game of stacking academic credit just like any other scientist. Although the pursuit of a maximum h-index and the perversion of bibliometrics (Hicks et al. 2015) are still less pronounced in textual scholarship than in science, nevertheless scholarly editors must also account for their efforts and demonstrate their academic relevance. The appropriation of authorship here is pivotal because it is what scholarship and tenure in the humanities are primarily built upon. Within institutional registration systems monographs and peer reviewed journal articles are usually the top ranking indicators of research output. And it remains generally true that in the humanities, monographs – which supposedly represent more effort in authorship than any other type of publication – are far more valued than journal or special issue contributions (Bishop 2012). The single most common feature of these types of output is that they all consist of authored text. Thus for all practical purposes, the relevance, the scientific impact, or valorization of research effort in the humanities is equated with the authorship of published texts.

6.3 The Author and the Engineer

Scholarly editing is the appropriation of revisionary authorship. Its aim is to make a scholarly motivated contribution to the textual "archive" – which may be taken narrowly and concretely as a digital archive or edition of the text and all materials pertaining to it, or can be broadly understood as the more metaphysical human culture memory system that Jerome McGann implies. By performing the scholarly editing the editor becomes also an agent in a process that we may call the social construction of the interpretation of the text. Then, if we turn to digital scholarly editions, a question arises. Are software code, the engineering involved with its creation, and the programmer performing that engineering mere inertial elements in the process that is digital scholarly editing? In the discourse on scholarly editing the roles of code and the coder seem strikingly lacking. Jerome McGann makes mention of related skills and methods (McGann 2013:344) but mostly to address them as auxiliary technologies and to discard them as non-essential to scholarly editing. Must philology indeed be understood solely as the science of the memory of manuscript and print words? McGann writes about philology as "the fundamental science of human memory" – if anything, that is broadly inclusive. It is hard to imagine that such a science would not transcend paper and ink. On those grounds, it would seem that philology must not be constrained to the scholarly task of creating the "archive" of non-digital media. Why not argue that philology be understood as the memory of human semiotics? Would it not be far more interesting and intellectually challenging to resist retreating to a gated community of analogue sources, and to venture out in search of the commonalities and differences between the semiotics of the analogue and the digital? We need not arrive at some unified theory of philology, but certainly we want to understand those two cultures of textuality better through their differences by which they define each other. Let us start our exploration by asking a question congruent to the one which started the previous section: how does authorship relate to writing code? To answer that question we must also have a basic understanding at least of the nature of software code itself.

Any functioning programming language is in itself a semiotic system. It consists of symbols that denote a meaning by referring to objects and concepts. It has a syntax, and there is a pragmatics to its use. Most programming languages express themselves in the form of text, usually as a mixture of textual, logical, and mathematical symbols. Because they invoke meaning just like the symbols of human language the use of these symbols is not restricted to

```
<Script Language="...">

<!-- Begin

function V_____() {
   var character = V_____.fe|ma[n]le;
   V_____ = x.disorder(0, 1);
   if(V_____ >= psycho)
       multiPersonality();
   var isV_____ = (V_____.fe|ma[n]le.introduce("V_____")! == 1);
   var persona = new self("Mother without eyes & nose", "Teacher who always
   yells at her pupils");
   a = self;
   b = mother + teacher;
}

function multiPersonality() {
   self = self[a];
   aa = "<action='+personalities+'>murder women who have big mouths and
   loud voices; V_____ is both the mother & teacher</action>";
   if(isV_____) {
       mother's big mouth.cursing.everything(all day);
       teacher's voice & rules.scolding.nonstop();
   }
   else
       hatred.innerV_____ = afraid of them;
   if (a < b || a > b) b = -1;
   if (a==1) b- = 1;
   if (a==0) b+ = 1;
   xx = setTimeOut("multiPersonality()", alternative);
}

//End -->

</Script>
```

Figure 6.1: Example of "codework" (Thuan 2000; Ciccoricco 2010)

.. my.time: my time: it _c(wh)or(e)por(ous+h)ate_ _experience____he(u)rtz___.] [end]

Figure 6.2: An example of Mezangelle (Myers 2012).

the realm of computing. Codework, for instance, is a poetic form that mixes natural language and programming language elements (Raley 2002). It thus creates meaning from intermingling natural language semantics and procedural elements, such as "go" and "next", from code (cf. figure 6.1). Mezangelle, a poetic-artistic language developed in the 1990s by Australia-based Internet artist Mez Breeze (an alter ego of Mary-Anne Breeze) is another example of a hybrid computer and human language, based primarily on portmanteau words derived from code and human language vocabulary (Raley 2002, cf. figure 6.2).

But also by itself code in a programming language expresses meaning. And just like "normal" text it expresses meaning on multiple levels. There is a literal level. A simple line of code in Ruby – one of the more recent popular general-purpose programming languages – may read:

```
puts( 'Hello World!' )
```

The literal meaning that will be obvious to a (Ruby) software engineer is that this one line program will print the words "Hello World!" to some terminal (usually some output medium such as the computer's screen). There is a conceptual meaning to writing code as well, which is mostly tied to what engineers tend to call "the model". The model is comprised of those parts of the code that express the concepts and relations of the real world phenomena that the engineer is trying to encapsulate and express with her code. To give an example, a very simple "dog model" may for instance look like this in Ruby:

```
class Dog
  def initialize( name_of_dog ) @name = name_of_dog
end

def bark
  puts( 'Bark!' ) end
end
```

An engineer re-using this code is able to instantiate a new Ruby-dog by typing and executing `my_dog = Dog.new( 'Skippy' )`. She can then make it "bark" by calling out to the function `bark` – like this: `my_dog.bark` – which will result in the word "Bark!" appearing on the screen. And, apart from these conceptual and literal meanings of code, there is a pragmatic meaning to computer code as well. In the case of the Hello World! example, this specific use of the words "hello" and "world" signals to almost any engineer that here is a code example intended to get a novice underway with programming in that language.

Thus, understood as the language that expresses software, code has a semiotic nature, simply because it is a symbolic means to an end. Code is also essentially a series of symbols that instruct a computer to perform certain actions. As a result, program code has a dual realization. One of these is the code itself, as text, which carries forth its "operative" meaning for a human interpreter. That is, those more or less fluent in that particular programming language can gauge how the programmer intended the program to operate. This realization of code is different from the actual result of executing the program, which is its performance. That performance might be a particular calculation result, an interface for interaction, a visualization, any combination of those, a text, and so forth. The first realization of code, code-as-text, enables a programmer to have the code produce almost any other semiotic expression, including expressions of the same or other computer languages, as its second realization: the execution result.

Code creates a multi-dimensional space of structure and meaning, just as text does (Huitfeldt 1995). Following Compagnon as in the previous section, a text may have contemporary meanings and anachronistic ones; it will keep on creating new meanings over time, as long as it is read. Programmers re-use code. They re-use their own and they re-use code that they find in "code libraries" (programming idiom is sometimes pretty self-explanatory) of other programmers. Barthes' observation, that a text is not a line of words releasing a single meaning but a multi-dimensional space which recombines a variety of writings (Barthes 1967), is as true for code as it is for text. In a similar vein Friedrich Kittler calls on Derrida (Kittler 1993a:225–226) to argue that the creation of the digital form of text that code is, is indeed nothing more or less than the continuation of writing with different means. But he also points to the essential difference between the two sorts of writing: that digital writing is "im Unterschied zu allen Schreibwerkzeugen der Geschichte auch imstande [...] selber zu lesen und zu schreiben" (Kittler 1993a:226). Differently put, the difference between text and code is that of the difference between performance and auto-performance. A print text cannot itself perform, it needs a reader to accomplish that. Digital text, code, can perform. It can even self-perform, though this is seldom the technical reality; in the vast majority of cases it is read and executed by another piece of software that follows its instructions, much indeed like a Jacquard loom (Ceruzzi 2012). Donald Knuth realized that these different natures of code as on the one hand machine instructions and the other hand meaningful text mixed two types of literacies: the literacy of the executable code and the literacy of non-executable text. This led him to develop a computer system that explicitly caters to these different literacies, although his direct motivation was pragmatic: "I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: 'Literate Programming' " (Knuth 1984:97).

Knuth's WEB system was aimed at expressing a program as a "web of ideas [...] in a natural and satisfying way". Therefore the source text, written by a programmer, would be interpreted by his WEB system which as the result of a choice would produce either a computer program or a narrative explaining the concept, ideas, aims, and functioning of that program. Knuth's system is a sensible solution to explicate the non self-explanatory parts of code. In its narrative expression it clarifies the intent of the program, or for that matter that which the programmer wants to clarify about his or her intent. It is therefore very similar to the accountability that scholarly editors usually attend to, in introductions or other paratext of editions, explaining their editorial principles and practices.

Of course the ability of code to read and write code then again makes program code rather different from "normal" text. Both kinds of text have the ability to encapsulate each other. For instance, in the case of `puts( 'Hello World!' )` the ruby code has been encapsulated by the text of this very paragraph. The text "Hello World!" within that code is an example of code encapsulating "normal" text. But the ability of code to operate on text is not symmetrical. Code can operate on text, which results in the same, new, or transformed text:

```
'Svbstitvte'.gsub( 'v', 'u' )
```

The result of the above code is the word "Substitute". Text cannot operate on code in the same way: encapsulating code within text, or inserting arbitrary text into code cripples the code's ability to execute. Code can operate on text without changing the textual nature of text. However, if text operates on code, it changes the nature of code to that of text. The keyword here is "operate". A text by itself does not operate on reality. It needs an agent (i.e. a human reader) to do so. Code in contrast, as Vitali-Rosati (2016) argues, does operate on reality and produces reality when executed. In this sense code has a proxied reproducible agency that can be congruent with the intent of its author. Obviously, as with the authorship of text, code may also be an imprecise translation of the intent of its creator. It is equally obvious that code, as with text, may result in vastly incongruent meanings and interpretations when it is read subjectively. The difference however with authorship – and thus also with the revisionary authorship of scholarly editing – is that code is a meticulous description of the exact operations that it performs. Whereas editing foremost produces a result and an inexact account of how it was reached, code allows for the preservation of all editorial actions. Its ability is to provide an exact and reproducible provenance of the scholarly edition.

Code thus functions as a beefed-up version of text. It shares with text all the problems pertaining to meaning, interpretation, and intent. The aspect of performance, or reproducible action, suggests that code has in addition a certain agency, which may congruently – but much more likely incongruently – effect some of the coder's intent. The limits and effects of the politics and agency of objects like books and software is a discourse in itself (Woolgar and Cooper 1999). Vitali-Rosati (2016) and Berry (2014) seem to argue that these effects in the case of software are more immediate and concrete than in the case of text. Suffice to say here that this agency exists. And it suffices to conclude that code is indeed very similar to text – so much so that Marino

(2006) has proposed, as said (chapter 5), "that we should analyse and explicate code as a text like any other, 'a sign system with its own rhetoric' and cultural embeddedness".

The similarities between code and text then suggest that coding is similar to (scholarly) authorship as well. Indeed software development shares all the basic properties Foucault attributes to the "author-function". Foucault traces the roots of the concept of authorship back to the hermeneutics of Bible exegesis, and points to "transhistorical constants" that pertain to the creation of the "author-function". Four of these constants originate from St. Jerome's *De Viris Illustribus* (4th century CE); they can be summarized as consistency of quality, consistency of concept and theory, consistency of style, and authenticity. It is striking that these features are among the most highly debated properties of code and code authorship too. Extensive and often automated software testing frameworks have been developed in order to guard against a perceived lack of quality, against "cowboy coding" and "spaghetti code". "Design patterns"[2], "cookbooks", and "recipes"[3] have sprung up to share standard approaches to solving common problems, which may be compared to tropes or motifs in text authorship. Consistency of concept and theory translates directly to the quality of models designed and metaphors used for particular pieces of software, which are kept as consistent as possible during a software package's lifecycle. Fierce disputes are fought over computer languages and the coding style that ought to be used when expressing models in them. Quality and particular style are often discussed in ways that would easily be recognized by scholars of literature, such as when Yukihiro Matsumoto, the creator of the Ruby programming language, talks about quality, style, and elegance (Chandra 2014). In all of this code authorship seems not to differ greatly from the authorship of literary or scholarly texts.

I have argued above that scholarly editing is revisionary authorship. The authorship of code in the case of textual scholarship is also revisionary authorship. Its impact however is not limited to the text – it extends to the process of scholarship. Software rewrites scholarly editing. The most obvious example of this agency of code is visible in the shift of scholarly editing from an

[2] https://sourcemaking.com/design_patterns

[3] http://shop.oreilly.com/category/series/cookbooks.do

academic activity undertaken by individual scholars to a collaborative effort. Sahle (2013) and others have pointed to the teamwork that often characterizes digital scholarly editing. In stark contrast to the negatively stereotypical, yet in some cases still wonderfully fitting, image of the textual scholar working in the splendid isolation of an ivory tower, development of a high-end digital edition is often a collaboration involving computer engineers, graphical interface designers, an IT project manager, a data steward, and possibly several others with additional roles. In the categories of authorship that Love (2002) distinguishes, and whose concept of editing as revisionary authorship we have seen above, this would compel us to label such digital scholarly editing as collaborative revisionary authorship. Collaborations in textual scholarship are hardly new of course – one need only consider the Ulysses according to Gabler, which is actually the Ulysses according to Gabler, Steppe, and Melchior. However the software that enables so called computer-supported collaborative work (Greif 1988) created affordances that have certainly facilitated this type of collaborative editing to great extent.

These affordances are also a catalyst that extends the process of social construction of interpretation beyond the in-group of textual scholars. For instance, they enable the technological platforms that make the concept of open-ended or social editions feasible (Siemens et al. 2012; Causer and Terras 2014), and allow inclusion of crowdsourcing efforts into the practices of scholarly editing. This means that scholarly editors increasingly involve people who are not fully formally trained, such as students and interested amateurs, to participate in the editing process (Brumfield 2013). Web-based tools for instance make it easy and affordable to engage a wider audience in the transcription of manuscript material that can only be digitized by hand. In this way an important part of scholarly editing can be outsourced to a voluntary labour force. At present it seems that in most cases the use of "free labour" is as far as the openness of the editing process goes. However there is no technological bar to the idea that "all readers may become editors too" (Robinson 2004). Recalling for a moment the ability of code to keep an exact record of actions, a scholarly edition could be opened to annotation and reuse at large, with no threat to a base layer of text that has been deemed to possess a certain level of scholarly quality. Software thus enables us to rewrite scholarly editions as social sites of knowledge building.

These are examples of software's ability to rewrite scholarship on a macro level. However the rewriting of textual scholarship with code applies at a more fine grained level as well. In his introduction to his book *Textual Scholarship* David Greetham (1994) writes: "textual scholars study *process*". The italics are his; the point was that important to him. That in any case was most fortunate because, after all, computers excel at reproducing process. Digital scholarly editions – though admittedly most of these are documentary digital metaphors for the codex – allow scholars to add process and performance to texts. Work at the Huygens Institute for the History of the Netherlands can be pointed to as an example. This institute started an open source project that resulted in an international community of developers who work continuously on a piece of software called CollateX. This is a collation engine that delegates the scholarly task of aligning variant texts to an algorithm based on a combination of computation and scholarly heuristics (Haentjens Dekker et al. 2015). Graph models are then used for a precise and computational description of the differences between texts (see figure 6.3 for a small example). Put differently: these graph models describe to the highest precision the textual variation between different witness texts of the same work, in a more exhaustive manner than is feasible through written argument or printed visualization. Graph models by themselves are not new.[4] However, their application in textual scholarship is still pioneering work (e.g. Schmidt and Colomb 2009, Haentjens Dekker and Birnbaum 2017). They represent a technology-based advancement of textual scholarship methodology. To argue for and implement their application therefore qualifies as a scholarly contribution. Yet the methodological research behind the application of these graph models is only partially the work of the textual scholars involved. Most of it is done by the software engineers who have acquired a scholarly understanding of the text-theoretical problem.

When such analytic software is applied many scholarly decisions about the text are delegated to the code itself. When used to produce a scholarly representation of a text, this means that the performance of the code becomes part of the revisionary authorship that scholarly editing is. This in turn means that the authority over the claim to a textual truth shifts and redistributes; it is partially, albeit implicitly, claimed by the engineers of the software. To

[4] Cf. http://www.textgraphs.org/

an extent the engineers become accomplices to revisionary authorship and agents in the social construction of interpretation.

Coding work is often not recognized as an integral scholarly part of the scholarly process. It is mostly seen as a service, as the offshoring or outsourcing of production work. We see here however that the creation of code that produces data and interpretation involves scholarly decisions and thus enlarges the network of agents that have pertinence to the interpretation of a text.

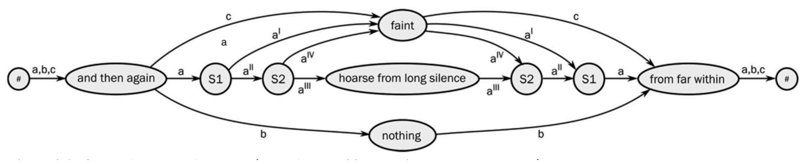


Figure 6.3: Graph model of genetic stages in a text (Haentjens Dekker et al. 2015).

Bernard Cerquiglini in "In Praise of the Variant: A Critical History of Philology" argues that any edition of a text is not that text itself, but a theory, an argument about that text (Cerquiglini 1999). This circumvents the methodological conundrum of authorial intent and subjective interpretation. In this conception textual scholarship no longer claims to know how a certain text should be interpreted. Rather, it operates from the premise that there exist hypotheses about the interpretation of a text, and that these hypotheses can be expressed as scholarly editions. Such a perspective relates textual criticism and literary criticism by stressing their interpretative stance, effectively reversing that relationship. The edition is no longer the base layer of philological fact – that "impossible truth" – but an expression of a process of critical interpretation to which multiple agents, human and technological, contribute.

Also software and its creators now become part of this interpretation. Bauer (2011) in answering a recurring question on the perceived lack of theory in coding and digital humanities practice quotes Susan Smulyan calling out: "The database is the theory! This is real theoretical work!" Galey and Ruecker (2010) have argued similarly that design, prototypes, and code can be scholarly arguments as well. Further examples – such as the understanding of coding as a form of disciplined play and in that sense as an instrument of research (Rockwell 2003), and reflections on the hermeneutic role of code in the digital humanities (Ramsay 2011b) – support the view of coding as a form of scholarly authorship and argument as well.

But if digital objects and code are an integral part of the scholarly argument then their producers have an obligation to claim the contribution they make to it – not just to make a righteous claim to academic credit, but foremost in order to be accountable for the scholarly argument they co-create. The intent of the revisionary authorship that code produces as its reality is not self-evident nor self-explanatory. This is true for any form of authorship, as we have seen. Indeed this was why Donald Knuth came up with the concept of literate programming. At the very least programmers should claim explicitly the revisionary authorship they appropriate through executable code. This is not simply a claim to the authorship of the code. The responsible use of repositories such as GitHub[5] will more or less automatically track who authored what in which version – again code delivers an exact provenance. Rather, my assertion here is that programmers need to claim explicitly the collaborative authorship in which their code is an agent and which results in scholarly texts or digital scholarly editions.

Such a claim is rarely, if ever, made. One reason for this absence is the legal implications such claims would undoubtedly have in the case of commercial software. However, in the case of scholarship it is not a legal issue, but a question of scholarly accountability, responsibility, and value which concerns us. Another reason that in part may explain the lack of claiming authorship might arguably be the somewhat ambivalent terminology around "authorship" in the realm of programming. Authors of software do not often talk about coding in terms normally associated with text authoring. Rather, the jargon has associations with engineering and technology. One talks about "coding" or "hacking", about "building" or "developing" software. Creators of software code talk about themselves as "software engineers" or "developers", and at a higher level of responsibility perhaps as "lead developer" or "architect", even if their methods may be less rigorous than such labels suggest (Bogost 2015). And indeed one reason is likely to be that not all scholars

[5]https://github.com/

recognize or wish to acknowledge code as a scholarly object (Schreibman et al. 2011). Yet, it should be claimed as such. Christine Borgman in a recent publication *Big Data, Little Data, No Data: Scholarship in the Networked World* argues that the simplest solution to this problem of crediting the scientific effort associated with coding and data curation is to list software engineers and data stewards, along with all others involved, as contributors – much like the credit roll of a movie (Borgman 2015). This does occasionally happen (e.g. for the digital scholarly edition of Vincent van Gogh's letters[6]) but rather as the exception that proves the rule.

If claiming their scholarly authorship is hard, actually being accountable in a scholarly sense for the agency of their code might be even harder for programmers. In the domain of digital textual scholarship, the peer review of code is scarcely even a nascent concept. Frabetti (2012) has argued that to solve this problem a full "reconceptualization" of digital humanities is necessary. By this she means an understanding in terms other than the technical, such as predominates in computer science, and an understanding in terms other than that of new media studies that focuses predominantly on the "consumables" resulting from the digital, such as digital books, games, and so forth. According to Frabetti this requires "a close, even intimate, engagement with digitality and with software itself". The hard question Frabetti advances is: how can we apply scholarly criticism to code?

6.4 Being Critical About Code

The post-structuralist move of literary criticism and the resulting decentering and "death" of the author were in part a result of the influence of critical theory on literary criticism. In general critical theory questions, critiques, and challenges forms of authority. Thus, in literary criticism it led to questioning the authority of the author. In the last few years critical theory has also made some inroads into the critical examination of code. Richard Coyne (1995), Morozov (2013), Berry (2014), and many others call attention to the pervasive, but covert and non-neutral ways in which software affects society

[6]http://vangoghletters.org/vg/credits3.html

and humanistic expression. Code has the potential to change and rewrite the structure and rules of culture, but textual scholarship and digital humanities have hardly begun to establish an effective critical mode towards this softwarization, and moreover to its own softwarization. Ramsay (2011b) after being heavily criticized withdrew in part from his blunt claim that to belong to the digital humanities one needs to be able to author code. Yet I would indeed argue that it is pivotal that code literacy becomes to a certain extent an intrinsic part of humanities methodology. A discipline tasked with the critical approach of literature and culture but illiterate in the "native" language in which these cultural artifacts are now increasingly created is a methodologically defective discipline. The arguments of Morozov and Berry – and to a lesser extent that of Coyne as well – are admittedly politicized, pushing back on a "neoliberal" ideology. There is nothing wrong with such a deep political engagement. Textual scholarship does not operate in a vacuum. However, the political argument is not needed to warrant critical examination of software and code structures from a textual scholarship perspective. After all, as we have seen, there are very good scholarly motivations to critically examine code. We should understand critical theory not in its guise of politicized agent, but as the pragmatic intellectual catalyst of liberation of thought and perspectives. Moreover, as a "simple" matter of intellectual obligation to its methodology, textual scholarship should find enough cause to critically examine the structures that computational methods and digitization push upon it, and if necessary should push back upon them with force of argument. Adopted uncritically these structures can be narrowly normative. Landow (2006:59) refers to Tom Meyer's explanation of this problem by pointing out that humans generally rely on "arborescent" structures – binary thought, genealogies, hierarchies, and divisions that dissect the overwhelming amount of information that is available to us into more easily assimilable bits. The risk is that these structures for knowledge organization become the only methods of understanding, in which case they "limit instead of enhance or liberate our thought". It is paramount therefore that we examine such structures critically. Also when they are digital or exist as code, as digital information and code are not, in fact, neutral.

Interestingly then, it is Jerome McGann who has provided a very good example of such critical thought. In his "Marking Texts of Many Dimensions"

(McGann 2004), he considers literature as an autopoietic system – a system that rewrites itself continuously, and that perpetually self-organizes and self-re-organizes the same information to change and add meaning and interpretation. Hierarchical information structures, such as XML (which exerts a considerable push on textual scholarship's methodology), are inadequate to describe and model systems of such nature. This is firstly because such markup hierarchies tend to be intentionally static. They are generally not meant to be changed but to function as the lasting expression of some argument about a text resulting from textual criticism. Secondly, such structures are unable to represent multiple perspectives on the same phenomena, because to do so would result in multiple hierarchies which cannot be represented concurrently. This is called the "problem of overlap" in digital textual scholarship. Conjecturing the autopoietic nature of literature allows McGann to critically attack Sperberg-McQueen's summation of the problem of overlap in hierarchical markup structures as being a residual problem (Sperberg-McQueen 2002): "But those matters are not residual, they are the hem of a quantum garment. [...] No autopoietic process or form can be simulated under the horizon of a structural model like SGML, not even topic maps" (McGann 2004).

Of more interest here than the technical argument is that McGann is employing a process of critical thought towards an information technology solution. Based on critical examination he is questioning the authority of that solution. Reasoning from the perspective of textual and literary criticism he argues the inadequacy of the "textual ideology" that underpins hierarchical markup languages – a technology that is all too often unquestioningly taken as neutral and adequate.

6.5 Conclusion

Scholarly editing and the creation of code that is part of that process are both forms of revisionary authorship that contribute through scholarly argument to a hypothetical expression of a text. It would therefore be academically irresponsible not to develop a methodology that can acknowledge the scholarly contribution and status of the creation of digital objects and code, and that enables scholarship to perform systematic critical examination of these objects and that code. Although relevant theory and techniques abound in both the scholarly and technological domain, there is not yet a theoretical or methodological framework that combines them into a method for critiquing scholarly code. Exploring possible methods for criticism of scholarly code is a task that must be taken seriously within digital textual scholarship and digital humanities if their practitioners care about scholarly rigor. The outline of what such criticism could look like was discussed in the previous chapter. The salient observation here is that if we would not explore and construct such a framework, we implicitly make the false assertion that code is an inert agent – code and its authors are neither neutral nor rigorous simply because code is technology. The so-called Postmodernist Generator (Bulhak 1996) is an excellent example of this.[7] This piece of code will generate sentences like: "Lacan uses the term 'prematerialist desublimation' to denote a self-justifying paradox". The sentence generation is based on an explicit grammar modeling the text of a real postmodernist reader (to wit, Natoli 1993). The professed intent of the code is wonderfully self-defeating. It claims to substantiate the meaninglessness of postmodern jargon, but its source reveals unscientific bias: "we'll be ontologically masturbating in relation to the works of various artists and 'artists' a lot...".[8] It claims to produce meaningless text, but rather convincingly demonstrates the post-structuralist point made by Barthes (1967) that the origin of meaning lies exclusively in language itself.

[7] http://www.elsewhere.org/journal/pomo/

[8] http://runme.org/feature/read/+postmodgen/+57

Chapter 7

Apparatus vs. Graph: New Models and Interfaces for Text[1]

> "Interfaces are not simply objects or boundary points. They are autonomous zones of activity. Interfaces are not things, but rather processes that effect a result of whatever kind. For this reason I will be speaking not so much about particular interface objects (screens, keyboards), but interface effects. And in speaking about them I will not be satisfied just to say an interface is defined in such and such a way, but to show how it exists that way for specific social and historical reasons."
>
> (Galloway 2012)

7.1 Introduction

It is often said that textual scholarship is in transition (Kirschenbaum 2012), provoked by the increasing digitalization of society and culture. Yet perhaps textual scholarship itself is not so much in transition as is the basic "fabric" of its object of study, text. Almost all texts that are created today have at

[1] A previous version of this chapter has appeared as Van Zundert, Joris J., and Tara L. Andrews. 2016. "Apparatus vs. Graph: New Models and Interfaces for Text." In *Interface Critique*, edited by Florian Hadler and Joachim Haupt, 139:183–206. Kaleidogramme. Berlin: Kulturverlag Kadmos. The article is co-authored as a result of genuine collaborative thinking and research. Prof. Dr. Andrews provided most of the writing in the latter part of section 7.3 and most of the writing in section 7.5.

some point been produced or processed by a computer system, and hence almost all text today have also a digital existence, even if the final presentation is reliant on some physical medium. The field of scholarly editing has been struggling for over three decades to come to terms with how it could or should treat digital text, and to seek new contact points between the digital and the scholarly in the realm of text (cf. e.g. Landow 1994, Shillingsburg 2006; Robinson 2013a). These points of contact all rely on interfaces that allow textual scholars to interact with digital text. Any digitally informed transition to appear in textual scholarship should thus be tied to changing practices resulting from textual scholars interacting with texts by means of these interfaces. In this chapter a novel scholarly interface to text is presented that was developed in the course of research and development work by me and Tara L. Andrews. It will be argued that this interface presents text in a way distinctly different from the interfaces more commonly used in digital textual scholarship. The difference arises from applying a model for text that is at a remove from the well-known book or codex model. Arguably textual scholarship could enhance its engagement with text and its research capabilities by stepping back from the prevailing book model, and applying a more diverse range of models and interfaces.

7.2 The Computational Status Quo: Text-as-Sequence

The invention and development of the universal computer opened the way for the computational processing of textual information (Davis 2012). Although it is something of a foundation myth of digital humanities, the computationally aggregated concordance *Index Thomisticus*, a herculean effort by Father Roberto Busa first planned in 1946 with the cooperation of IBM and continuing into the 1970s, is usually seen as the first tangible result of textual computationality in the humanities (Winter 1999). Computational tractability of text has delivered powerful new means of curation and analysis to the hands of textual scholars. In textual analysis, stylometrics and stemmatology in particular were early "success stories" of computational approaches. Stylometrics is the – usually quantitative or statistically underpinned – analysis of literary style; its early applications included the identification of the author of a text based on its stylistic characteristics, and the field has expanded to investigate questions of genre, topic clustering, and narrative sequence. Stemmatology is the application of various methods, some based on statistics and others on deductive reasoning, to infer the order in which manuscript or print copies were derived from each other – somewhat akin to reconstructing a family tree of documents.

Our own earlier work provides examples of each. A stylometric measure known as Delta (Burrows 2002) was used in a case where two authors – one Penninc and one Vostaert, about neither of whom much is known – wrote successive portions of a medieval Dutch text, an Arthurian novel called *Walewein* (a Dutch analogue to the English "Gawain"). For reasons unknown Penninc did not finish work on the novel, and Vostaert later took it up, but there is no clear indication in the surviving text of where this transition occurred. Delta is effectively a measure for the use of high-frequency words. These are typically function words, also called functors in linguistics – generally small words with little lexical meaning, such as "is", "the", "me", and so on. The distribution pattern of this high frequency vocabulary is distinct for an individual author, and so acts as a sort of fingerprint. Applying an adaptation of this measure – which has become known as "rolling Delta" – to the *Walewein*, Van Dalen-Oskam and Van Zundert (2007) were able to get positive verification of the "handover" from Penninc to Vostaert (cf. figure 7.1). This type of research was only possible because of the application of computational statistics. Although the result could in theory have been derived by hand calculation, it would have been the work of many years to produce the millions of calculations needed.

The other example arises in the field of text stemmatology – the science of inferring a stemma (< Greek στέμματα, "pedigrees") of the copying relationship between manuscripts of an ancient or medieval text, based on evidence that can be found in the manuscripts available now. A "stemmatic method" bearing the name of the Classical and Germanic scholar Karl Lachmann was developed and refined over the course of the nineteenth century, and remains the standard for many textual scholars (Timpanaro 2005). Its use requires a collation of all available manuscript texts – a meticulous and formal detailed comparison of the text in its different forms – and depends on the

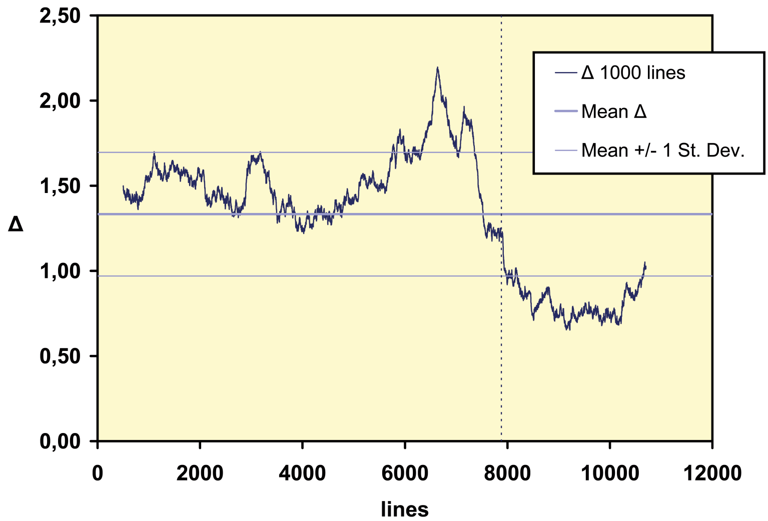


Figure 7.1: "Delta graph" showing changes in high frequency vocabulary through the text of Walewein, reproduced from Van Dalen-Oskam and Van Zundert 2007.

ability of the scholar to identify clear instances of copyist error that are unlikely to have been made twice.

Many textual scholars have argued that the Lachmannian method provides too much leeway for scholars to impart bias as to what is a "significant error" and how the stemma should thus be constructed. The method also runs into difficulty in the case of manuscript copies that were made by consulting more than one parent copy. Since the middle of the twentieth century the theory of Lachmann's method has been refined (Trovato 2014), but a competing methodology drawn from evolutionary biology has also been developed. This method, known as "cladistics", uses computational statistical methods to derive a genealogical tree according to the variants that are shared between subsets of manuscripts, with no need to distinguish scribal error in advance. Although cladistic methods arguably lend themselves more easily to computation, there have emerged in recent years a number of computational tools for creating stemmata semi-automatically according to Lachmannian principles. Examples include Roelli and Bachmann (2010), Camps and Cafiero (2014), and Barabucci, Di Iorio, and Vitali (2014). Many editors will employ a combination of methods to construct their stemmata; even so there remains some dispute in the field concerning which approach is more correct. In 2010 the Tree of Texts project began; its aim was to subject different methods of stemma creation to some form of empirical analysis. One result was a suite of tools for stemmatic analysis known as Stemmaweb.[2] The development of Stemmaweb required the creation of complex abstract computational models to represent text, its variation across manuscripts, and the "genetic" copying relation of those manuscripts to each other.

7.3 Function Follows Form: The Materiality of Text

Such computational analyses of texts and the statistical and digital tools used to perform them can be seen as new forms or modes of reading – "distant reading" as Franco Moretti (2013) calls it, as opposed to the customary "close reading". They can therefore be seen as new interfaces that allow readers

[2]https://stemmaweb.net/

and researchers to interact with text. Indeed Alexander Galloway defines interfaces not so much as objects or boundary points, but foremost as "processes that effect a result" (Galloway 2012:vii). In this sense these new digital or computational engagements with text are interfaces too: processes that allow a researcher to interact with text in a different mode, through a different process than by "reading the book". However, print text and the book, or codex, are to be regarded as interfaces by themselves as well. In everyday practice a text is identified with the book or another physical medium – journal, paper, etc. – carrying it, but in fact a book is just another way of engaging with the text. Even the text, understood as the semiotic system present on the pages, is just an interface. The actual text itself does not exist until and unless it is performed in some manner by an interpreter. Obviously the codex, especially in its function as a scholarly textual edition, has been an interface of paramount importance to the scholarly textual tradition.

Galloway argues that any interface that exists, exists for particular social and historical reasons. This is certainly the case with the scholarly print edition, where the use of the codex in textual scholarship and in scholarly print publications has been substantially shaped by the materiality of the page and the book. Although there exists a "language essentialist" line of reasoning, voiced by Thomas Tanselle and still widely adhered to, which holds that a text can be equated to its words, we also find scholars such as Jerome McGann who argue that the materiality of literature is an essential aspect of its content (Welch 2010). Looking at William Blake's illuminated lyrical work (figure 7.2) for instance, it becomes difficult to reject the influence that materiality has over the performance, experience, and interpretation of a poem's text. Blake's poem is a visual experience that goes beyond the characters and words. The text itself is not typeset, but engraved; this allows the textual and visual elements to flow together and emphasize each other in ways impossible to achieve through conventional typesetting. Through the poem's material properties Blake brings textual and visual elements into a "dialogue" that provides a different aesthetic experience, and therefore perhaps also a different or enhanced interpretation of the work – an experience that cannot be reduced to standard movable type.

However, that is often exactly what happens when works are remediated in the digital realm. Figure 7.3 depicts the remediation of the same poem

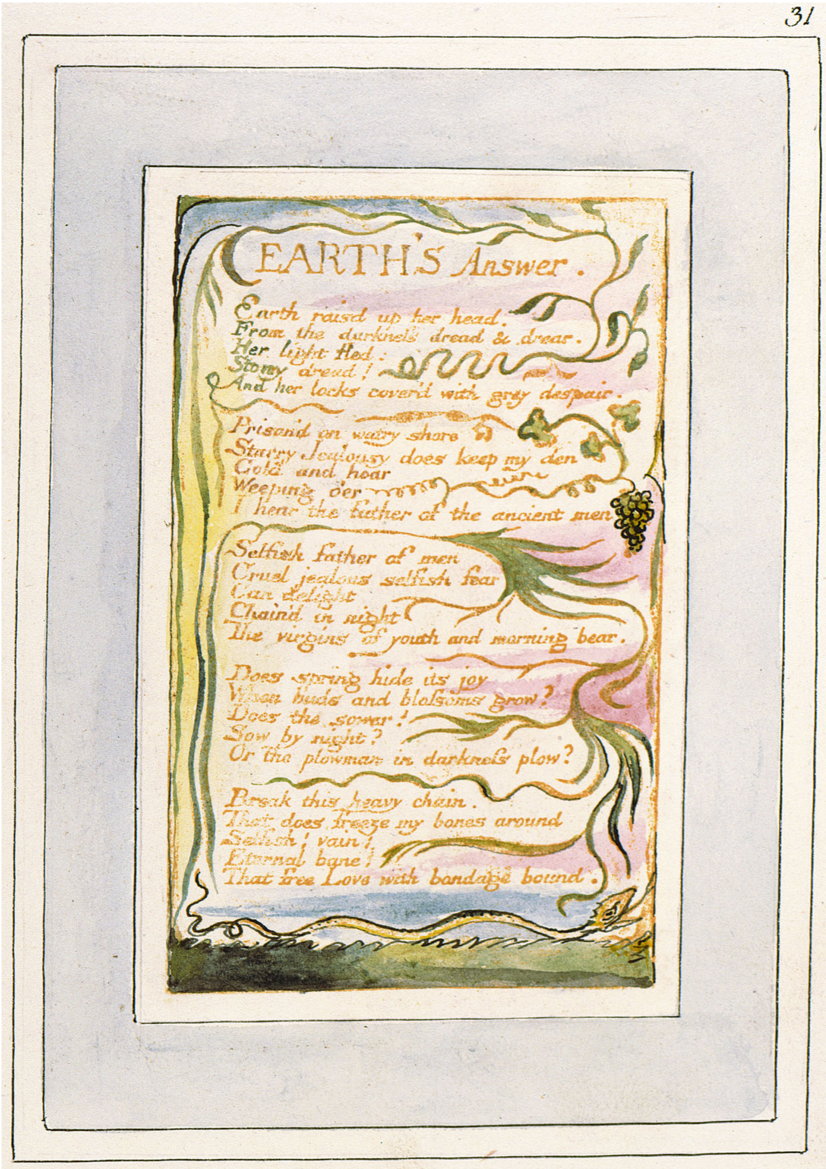
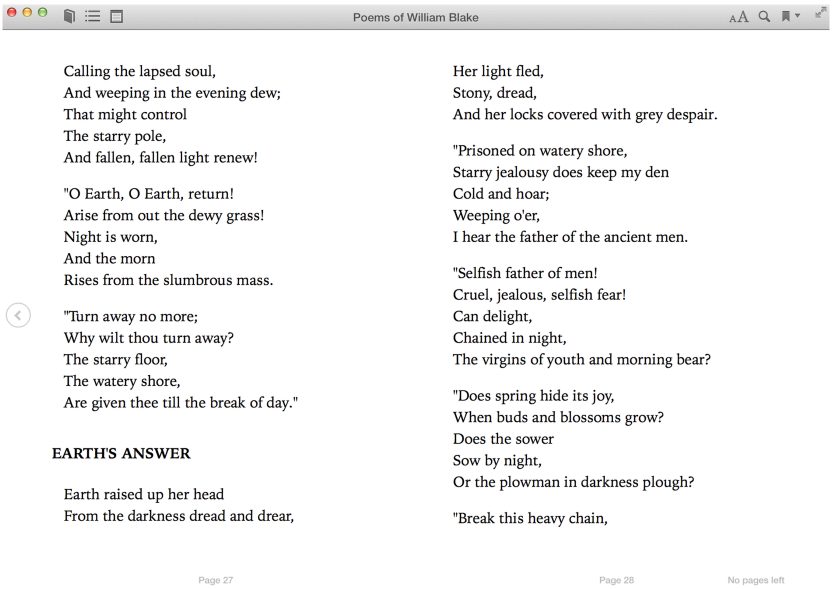


Figure 7.2: Songs of Innocence and of Experience, copy V, 1821 (Morgan Library and Museum) object 31 Earth's Answer (Public domain, via Wikimedia Commons).



Poems of William Blake

Calling the lapsed soul,
And weeping in the evening dew;
That might control
The starry pole,
And fallen, fallen light renew!

"O Earth, O Earth, return!
Arise from out the dewy grass!
Night is worn,
And the morn
Rises from the slumbrous mass.

"Turn away no more;
Why wilt thou turn away?
The starry floor,
The watery shore,
Are given thee till the break of day."

EARTH'S ANSWER

Earth raised up her head
From the darkness dread and drear,

Page 27

Her light fled,
Stony, dread,
And her locks covered with grey despair.

"Prisoned on watery shore,
Starry jealousy does keep my den
Cold and hoar;
Weeping o'er,
I hear the father of the ancient men.

"Selfish father of men!
Cruel, jealous, selfish fear!
Can delight,
Chained in night,
The virgins of youth and morning bear?

"Does spring hide its joy,
When buds and blossoms grow?
Does the sower
Sow by night,
Or the plowman in darkness plough?

"Break this heavy chain,

Page 28

No pages left

Figure 7.3: Poems of William Blake, p.27 of the Project Gutenberg eBook.

through one of the ePubs that are available via Project Gutenberg (Blake 1996[1821]). This version of the poem has what could be called the "Tansellean" essence of the text, which provides for a totally different experience, a distinctly different aesthetic. What makes this re-representation of Blake's text so different from its original is amongst others that it is offered as a machine readable text, which means that each glyph in the text is encoded as a number. Each number maps by human agreement – in this case the Unicode standard – to a character in an alphabet that is meaningful to human interpreters. It is this encoding which makes any letter "a" formally comparable to any other "a" in the computational or digital realm. This formalization of characters is what makes text computationally tractable, but it also causes the machine readable character to be reductive. The distinct material features that make Blake's poem unique are reduced to computationally comparable features. Text in the computational realm is likewise generally reduced to a "string": a linear series of characters and words that enabled – or rather were necessitated by – Morse code, telegraphy, Shannon's information theory and the Turing model (Shannon 1948; Davis 2012). This linear encoding was arguably reinforced and amplified by the severely limited memory and storage capabilities in the early days of computing. This formalism of linear information representation is both very powerful and very lossy: powerful because it allows for such stylometric and stemmatic analyses as we have shown above, but lossy with respect to the distinctive materiality that is characteristic of many real-world print texts. Yet as we have seen, the materiality of text is essential to textual scholarship and especially to scholarly editing, as it is directly connected to modes and possibilities of interpretation. Editions can only ignore the dialogue between the text under consideration and the materiality of its medium at the peril of reducing the "interpretational space" the original rendering of a text provides for.

7.4 Evolution of Scholarship in the Margins

There is a strong self-reflective relation between the materiality of the book and textual scholarship. The very materiality of the manuscript codex, and later the print publication, have shaped the formalisms of textual scholarship, and thus also in part the ways texts and documents are studied within textual scholarship. This can be gauged from a ninth century manuscript as depicted in figure 7.4. This page shows the early traces of how scholarship formalized itself – quite literally – in the margins of manuscript documents (Teeuwen 2011). The main text, supposedly a faithful rendering of an original text of Martianus Capella, forms the main "body" of the text on the page. Ninth-century scholars used the margin of the codex to squeeze in as many additional annotations, additional knowledge, and clarifications, as the space on the vellum would allow. Glosses were written between the lines, and Tironian notes (an ancient and medieval shorthand system for professional scribes) were used to compress even more information into the document. All these notes, annotations, glosses, and clarifications allowed the scholars to network their knowledge, to allow them to testify to the interaction between their thinking and the writing of their predecessor Martianus Capella (and, for that matter, the thinking of the various commentators before them).

The materiality of the medium – vellum and manuscript – has been pivotal in shaping how this dialectic and discourse took to paper. The margin was not invented especially for note taking, but it was there and came to be used for that purpose. Eventually the tradition of adding notes in the margins became more and more formalized and by the nineteenth century had evolved into what we now call "the apparatus" – a complex piece of scholarly technology to express as much scientific information about the text as possible in a confined space. Figure 7.5 shows a page from a typical modern critical edition of a classical or medieval text. The purpose of a critical edition, in most cases, is to present what, in the editor's judgment, is the most correct version of the text that can be deduced. In order to do this, the editor must collect as many manuscript copies of the text as can be found, compare them word by word to discover the differences between them, perform a stemmatic analysis to understand the approximate history of the text transmission, and finally construct a text based on all this evidence that is most closely in line with the editorial principles that the editor uses. In the critical edition, that text is presented at the top of the page, usually with line numbers for reference. The remainder of the page is taken up with a set of specially formatted footnotes (the apparatus) that encode, ideally, all of the information that the editor

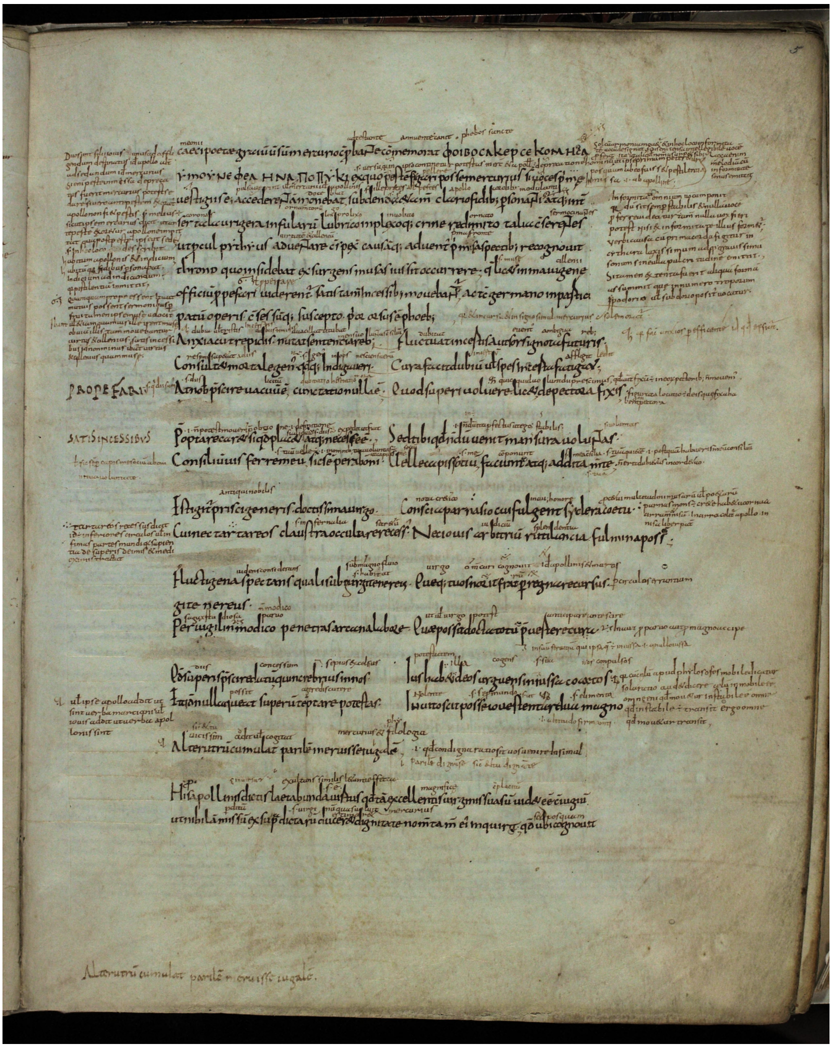


Figure 7.4: Leiden, University Library, Voss.Lat.F. 48, fol.5r.

τοῦ πνεύματος ἠκριβωκότων, τέσσαρας εἶναι τρόπους δι' ὧν συγχώρησις γίνεται ἁμαρτημάτων, δύο ἐνταῦθα, καὶ δύο ἐν τῷ μέλλοντι. Ἐπειδὴ οὐκ ἐξικνεῖται ἡ μνήμη ὅλου τοῦ χρόνου μνημονεύειν τὰ σφάλματα ἵνα ὑπὲρ αὐτῶν μετανοήσῃ ὁ ἄνθρωπος ἐνταῦθα, ᾠκονόμησε φιλάνθρωπος ὢν ὁ δεσπότης τῆς φύσεως καὶ ἡμῶν μὴ μετανοούντων τρόπους μετανοίας· ἐν μὲν τῷ μέλλοντι ὡς εἴρηται δύο· ὅταν τις ἀδιαφόρως ἐνταῦθα ἁμαρτήσας καὶ πάλιν ἀδιαφόρως ἀγαθοεργήσας, εἴτε εἰς οἶκτον καὶ συμπάθειαν πρὸς τὸν πλησίον κινηθεὶς ἢ ὅσα ἄλλα φιλανθρωπίας ἔχομεν, ταῦτα ἐν τῷ μέλλοντι αἰῶνι ἐν τῷ καιρῷ τῆς κρίσεως ζυγοστατούμενα ἐφ' ἃ τὴν ῥοπὴν ἕξει, συγχώρησις γίνεται· οὗτος μὲν ὁ εἷς τρόπος· ὁ δὲ δεύτερος ἐστὶν οὗτος· ὅτάν τις ἐν ἁμαρτίαις ἐνεχόμενος, ἀκούων δὲ τοῦ κυρίου λέγοντος *μὴ κρίνετε καὶ οὐ μὴ κριθήσεσθε*, φοβούμενος οὐδένα κρίνῃ, ἐν τῇ ἐξετάσει τῶν βεβιωμένων ὡς φύλαξ τῆς ἐντολῆς οὐ κρίνεται· τῆς γὰρ ἑαυτοῦ ἐντολῆς, οὐκ ἐπιλήσμων ὁ ἀψευδέστατος. Οἱ δὲ ἕτεροι δύο τρόποι ἐνταῦθα τὴν συγχώρησιν ἔχουσιν· ὅταν ἐν ἁμαρτίαις τίς ὢν οἰκονομῆται ἐκ τῆς προνοίας

ACPSTDEFHKQ ab ἐξετάσει (l. 22) ACPSTDEFGHQ

1.21–22 Matth. 7.1; Luc. 6.37

1.7 ἠκριβωκότων] ἠκριβηκότων PS **8** δι' ὧν…ἁμαρτημάτων] om. P **9** Ἐπειδὴ] γὰρ add. EKQ **11** μετανοήσῃ] μετανοήσει A **13–14** ὡς εἴρηται] om. DEFHKQ **14** ἐνταῦθα] om. T, post ἁμαρτήσας transp. Q **14–15** ἐνταῦθα…ἀδιαφόρως] om. P **14–15** ἁμαρτήσας…ἀγαθοεργήσας] ἀγαθοεργήσας ... ἁμαρτήσας T**17** ἔχομεν] ἔχωμεν A DF$^{a.c.}$, ἐχόμενα EKQ **20** τις] om. F**21** κρίνετε] κρίνεται A T DH (sed ε H$^{s.l.}$)**22** κριθήσεσθε] κριθήσεσθαι A, κριθῆτε Q |κρίνῃ] κρίνει A T DEFHKQ$^{a.c.}$**24** ἑαυτοῦ] om. PS**26** οἰκονομῆται] οἰκονομεῖται S T DEFGH

1.9 Ἐπειδὴ] γὰρ add. Max. **11** ἐνταῦθα] ante μετανοήσῃ transp. Max. **17** ἔχομεν] ἐχόμενα Max. **19–20** ὁ²… ἐστὶν] δεύτερος δὲ Max. **21–22** καὶ… κριθήσεσθε] ἵνα μὴ κριθῆτε Max. **22** κρίνῃ] κρίνει Max.

Figure 7.5: Extract of critical edition of Florilegium Coislinianum β′. This particular extract was taken from De Vos et al. (2010), it was rendered for illustration by T. Andrews using the Classical Text Editor software.

used to arrive at his or her conclusions.

This critical edition uses four separate apparatuses. In them, each manuscript that was used for the edition is represented by a sigil – an identifier such as A, B, C, etc. The first block holds the apparatus siglorum, which is a list of the manuscripts that were consulted for this particular span of text. In this case the reader can see from the apparatus that manuscript G was included beginning from the word ἐξετάσει on line 22. The second block notes references to the Bible; the fourth notes parallels with a different but related text. It is the third block, the apparatus criticus, where the majority of the editor's work is put on display. Each entry in this block gives a line number, a word that appears on that line of the text, and a list of the alternatives that appear in one or more manuscripts. Thus, for example, the first entry in the block

```
1. 7 ἠκριβωκότων] ἠκριβηκότων P S
```

tells the reader that, where ἠκριβωκότων appears on line 7 of this page, the manuscripts known as P and S read ἠκριβηκότων at that point instead. The next entry

```
8 δι’ ὧν...ἁμαρτημάτων] om. P
```

tells the reader that the text on line 8 from δι’ ὧν to ἁμαρτημάτων inclusive does not appear in (is omitted from, thus "om.") manuscript P. Abbreviations such as "om.", usually in Latin, appear fairly commonly in a critical apparatus, and the apparatus entries themselves can be arbitrarily complex in order to conserve space, which is at a premium in most printed texts. For example, beginning on the sixth line of the apparatus the reader encounters the entry

```
21 κρίνετε] κρίνεται A T DH (sed ε Hs.l.)
```

Here the reader is to understand that the word κρίνετε in line 21 appears in manuscripts A, T, D, and H as κρίνεται; that D and H are considered to be members of a related group of manuscripts (thus the lack of space between the sigla); and that H also contains the κρίνετε form written in above the line (s.l. = "supra lineam"). It is left to the reader to make the most probable interpretation of this fact – that the scribe of H initially wrote κρίνεται, and then corrected the text to read κρίνετε – although without either a more explicit indication or an image of the manuscript itself, the reader cannot be sure that this is what the editor meant to convey.

7.5 The Fear of Digital Freedom

The pair of examples above serves to show how the materiality of the codex – the technology of the book – shaped the practice and the technical apparatus that textual scholarship to this day applies in its publications, or in other words: how it shaped the interfaces textual scholarship creates to texts. There was in a certain sense a "dialectic" between materiality and intended purpose that mutually shaped the practice of scholarly commentary. Digital technology radically changes the nature of that dialectic. This is not, however, because digitality does away with materiality. At the most basic level, after all, bits of digital information are represented as material items: either as an analogue (electrical) signal within the most basic element of a computer's technical structure (the flip-flop, which is the simplest electronic switch architecture that can represent a dichotomy by keeping an electric current in either one of its two tiny circuits but not both), or in some more perpetuating material form (e.g. as tiny magnetized regions on a hard drive). As such there is little that is non-material about digital information (cf. for instance Kirschenbaum 2008). But what digital computation produced, propelled by the mathematical and technological developments by inter alia Turing and Shannon, were the most atomic elements of information, literally the bits, commonly understood as ones and the zeros. These most elementary atoms of computation allow us to derive by abstraction and composition any formal logic and language we can devise. The computer thus delivers an all but infinite freedom to create logic spaces and visual representations with and of these. One might regard computer code as an ultimate form of Lego: you can design your own blocks and even your own rules on how to stack them, unlimited and unbounded.

For interfaces and the creation of visualizations, this allows a radical freedom of expression. Suddenly anything that is imaginable as a visualization, as an interface for text within textual scholarship, may be realized. The limits of the material page, of the book, need no longer be the limits that guide the thinking about visualizing and interfacing with texts in textual scholarship. To a field such as textual scholarship, which – given one of its functions as the agent of preservation of texts – is consciously somewhat conservative in its methodology, this can actually be a disturbing development, especially in its function of philology. As Jerome McGann points out, philology is concerned with the memory of humanities, the record of humanistic expression – in what could be called almost a desperate attempt to stabilize that record (McGann 2013). Digitality in its radical freedom of expression and unlimited versatility of visual form, is obviously not a medium that allows for a convergence of method guided by materiality. This is not because of a perceived (but imaginary) loss of materiality, but rather because of the removal of the material limits of the page that influenced such a convergence of practice and expression. To a field that is concerned mostly with stabilizing its objects of study this may be more than a little disconcerting. In what could be a backlash in reaction to this radical freedom, some textual scholars seem to argue for a return and a reconfirmation within the digital environment of the structures that govern the book text, arguing that digitality should primarily provide a straightforward remediation of the book/text-structure in a digital environment, thus offering the digital text as a faithful simulacrum of the material and textual structures present in a material exemplar (Shillingsburg 2006; Pierazzo 2011). In which case the material limits of a physical medium are consciously superimposed onto a digital medium to limit its radical freedom.

Paradoxically then, digitality, which in principle suggests the liberation of material limits, currently serves primarily to reaffirm these limits. Moreover it seems even to narrow these limits and perhaps to reduce the expressiveness of textual materiality, as in the case of the visual poetry of William Blake and the inability of machine-readable text to capture and express its materiality.

However, rather than experiencing the radical freedom of the digital environment as a threat to the stabilizing functions of philology, we suggest that this freedom should be seen as an opportunity to explore the alignment of visualizations and interfaces of texts with the specific functions of a text. To support this we must return to an axiom in textual scholarship which holds that no text is actually stable. Rather, text is fluid. It is volatile: it gets changed, adapted, it crosses medium boundaries, it interacts with other texts, and so on (cf. for instance Bryant 2002; Coffee et al. 2012). Similarly an edition of a text, be it a scholarly one or not, is not to be equated with the actual text itself. An edition is always a re-illumination, a re-edition, a re-issue of a text. Due simply to the fact that a textual scholar had to interpret the text in order to produce it, the text is not the same as the text the scholar found in his or her witness(es). Because it has been interpreted, and has been given a new form by the textual scholar, it is by definition a new text. In the words of Bernard Cerquiglini (1999): an edition is not the text itself, an edition is an argument about that text. If a scholarly edition – or for that matter any edition – indeed is an argument about a text rather than the text itself, then it follows that the form should fit the argument. Of course this can mean that the edition should mimic as best as it can the "original" text, insofar as the scholarly editor chose to focus on the preservation aspect or conservational purpose of a digital re-rendering of the text. But utilizing the radical freedom of expression of the digital environment to the benefits and purposes of textual scholarship could also mean trying to fit a specialized interface to each of any perceived functions of the argument. Thus for instance, if the scholar is arguing the genesis of the text, the reader or user of the edition should be able to gauge the edition as if the text was being written: why indeed not animating or simulating the argued writing process, if the writing process is the aspect a scholar is scrutinizing? Then again, if the editor is arguing the reading of the text, we might simply want a clean reading text, bereft of any annotation or apparatus; perhaps a simple ePub text that can be read on a tablet. In other words an editor should be able and expected to create an interface fit for the purpose of the argument being set forth about the text.

Interfaces to text can in this sense offer different views on text, highlighting specific aspects that in traditional interfaces may be less pronounced or even impossible to express clearly. In computer science terms this is called many visualizations for one model (cf. for instance Gamma et al. 1994:14). In textual scholarship – admittedly abstracting away for a moment a number of text-philosophical subtleties – it can be called many instantiations or functions of a work. (For the relation between work and instantiation cf. Richard P. Smiraglia (2001:66); on the functions of text cf. Sahle (2013).) Deriving multiple views from one model will require something far more advanced than the string model however. The linear data structure of the string requires that too many additional features of a text be bound to it by other digital objects. Usually this results either in document markup schemes – such as XML[3] or Text Encoding Initiative's TEI-XML[4] – or an object model defined in programming code. Neither of these solutions is satisfying. Though an object model allows for maximal flexibility in expressing the structure, semantics, and commentary of a text, it lacks the attractive features of easy exchange and a data structure that concisely integrates all knowledge about a text, like a markup language such as XML does. Markup languages on the other hand lack the flexibility of compositional models and are unsuited to deal with multiple hierarchies, which means they are infeasible as a solution for expressing different views or differing interpretations of a text and its structures, as these are typically tied to or resulting in different markup hierarchies (cf. Sperberg-McQueen 2002; Schmidt 2014).

Between markup at one end of the solution space and object models at the other, graph structures seem ideally positioned as a middle ground. They do allow for expressing the complicated composite relations between objects that lend object models their power and versatility of modeling – and thus they are flexible and expressive enough to capture the multidimensional, multi-layered, composite relations between semiotics, structure, semantics, annotations, commentary, and various other layers of text. On the other hand they are integrated, self contained data structures much like XML hierarchies, easy to exchange and port for interoperability. In short we argue that graphs are ideally positioned to attenuate the reductive effect of basic linear computational strings, while avoiding some of the side effects of hierarchical markup structure.

[3]Exstensible Markup Language, see http://www.w3.org/XML/.
[4]http://tei-c.org/

Given these properties it is hardly surprising that graphs – as a data model – have found applications meanwhile in textual scholarship, most notably in the realm of describing and expressing textual variation (Schmidt and Colomb 2009; Haentjens Dekker et al. 2014). Indeed we use graphs as a model for text variation in our own work. However, we wondered if the graph itself could not serve as a far more intuitive and helpful "map" for variants description than the difficult to produce and difficult to use apparatus that we showed above. Thus we devised a visualization that depicts the graph itself to chart textual variation between different witnesses of the same text or work. We have also found this variant-specific visualization and interface to be a useful interactive tool for textual scholars.

7.6 Text and Its Variation as a Graph

The guiding principle of a variant graph is that all versions of the text have a beginning, and an end, and each version has a particular sequence of words that corresponds with the sequence of other versions. The simplest variant graph, of a single text, might look like the one depicted in figure 7.6 representing a popular sentence used for testing a typewriter.

Figure 7.6: Graph for a single text.

This fairly common sentence does have a common variation in popular culture that one might hear spoken. If we were to express that variation according to the rules of the printed apparatus criticus, we might say

```
dogs] sleeping dog *Oral trad*.
```

We can express that substitution in a graph as depicted in figure 7.7.

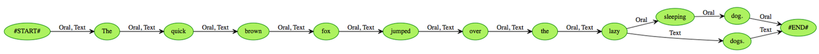


Figure 7.7: Substitution expressed in a graph.

Here our graph edges (the arrows) are labelled with the respective sources of our sentences – in this example, one comes from a written text ("Text") and the other from an oral tradition ("Oral"). Where the versions diverge, so too do the labels. By following the labels for any named version through the graph, the reader/user can reconstruct that version, unambiguously, in its entirety.

In the terminology of textual scholarship, we refer to each of the green nodes in the graph as a "reading". The labels on the edges refer to individual versions of the text, known as "witnesses". Readings that occur in corresponding ways in different witnesses, such as "dogs" versus "sleeping dog" in our example, are known as "variants", and it is these variants that the graph was developed to display. The variant graph is thus a means of expressing the relationships between several copies of the same text in a unified way, giving the reader an interface to the text that provides several immediate advantages over the apparatus criticus. The first is that the reader can quickly grasp the magnitude of the variance and fluidity of a text in a visually intuitive way. The second is that this interface refrains from inherently privileging any single version of the text; rather, the reader is free to consider the various textual copies as a collective entity, independent of any structure imposed by the author of the apparatus criticus.

The third advantage of the variant graph is that it can also be used to express more explicit information about the textual variation than is typically included in any printed edition – for example, the specific relationship that might exist between a pair of variants. To illustrate this, we can turn to the Greek text used in our apparatus criticus examples above. Our apparatus entry from line 7

```
7 ἠκριβωκότων] ἠκριβηκότων P S
```

becomes a simple division as depicted in the graph of figure 7.8.

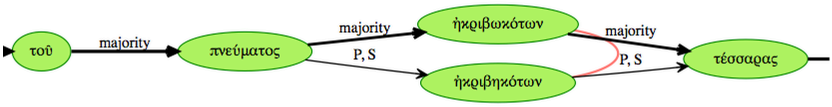


Figure 7.8: Simple division expressed in a graph.

with a red line to join the variant readings that, in this case, indicates that these are spelling variations of the same word. In this example we also see that the thickness of the lines in the text path is used to give an impression of the respective number of witnesses that follow each sequence.

The omission of the phrase recorded in our apparatus criticus for line 8 of the text

```
8 δι’ ὧν...ἁμαρτημάτων] om. P
```

can be represented in two ways. An initial graph might look as shown in figure 7.9.

Figure 7.9: Graph expressing an omission in witness P.

A textual scholar would most likely object to such a representation – she or he would argue that this graph shows the absence of five separate readings in P, suggesting five separate acts of omission by its scribe. A better graphical representation of the text might thus be constructed as in figure 7.10.

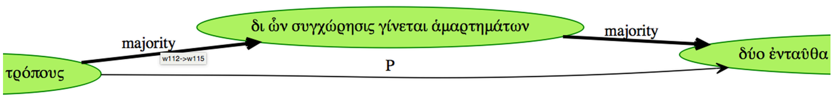


Figure 7.10: Graph expressing omission in witness P as singel reading.

In this case the variant is understood to be a single reading, comprising the entire phrase, that was omitted in a single act.

A more complex example from the original apparatus criticus is the following, which required three entries to describe in full.

```
14 ἐνταῦθα] om. T, post ἁμαρτήσας transp. Q
14–15 ἐνταῦθα...ἀδιαφόρως] om. P
14–15 ἁμαρτήσας...ἀγαθοεργήσας] ἀγαθοεργήσας ... ἁμαρτήσας T
```

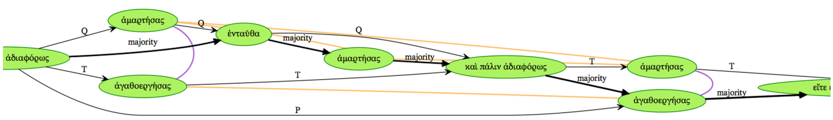


Figure 7.11: more complex variation in a graph.

Expressed as a graph (figure 7.11) the purple lines represent readings that have parallel grammatical structures, and the orange lines represent readings that are the same word, transposed to another location in the text. The three distinct apparatus criticus entries are thus unified into a single representation of the overall textual fluidity.

Although the graph interface to the text has the great advantage that no single version of the text need be prioritized, it is also possible to use color as an additional channel of information. As the editor studies a text, she or he may well wish to construct a "red thread" through the possible readings, marking out the argument to be made concerning the sequence of readings that might constitute the canonical text. This can be marked, quite literally, in red (figure 7.12).

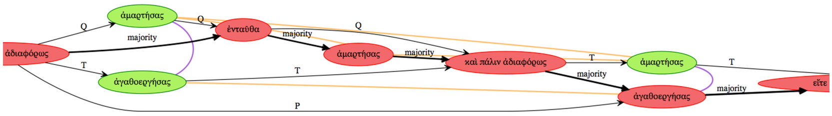


Figure 7.12: A 'red line' of edges marks a canonical reading.

7.7 Conclusion

Textual scholarship is rooted in a tradition where many scholarly functions demanded of a text were made part of that text within the material confines of the codex. Those material limits and physical context of the codex essentially converged with the conceptual demands of scholars to shape a highly technical and complex apparatus that combined the many functions that scholars wished to see represented in a text. The transmediation of text into the digital medium has so far been problematic from the perspective of textual scholarship, as the typical digital and computational forms have not provided a very welcoming environment for text. On the one hand, the basic and far too simplistic structure of the string severely denies the complexity of text. On the other hand, the radical freedom of computational modeling presents textual scholarship with a vastness of unexplored presentational possibilities that can be more terrifying than reassuring to a discipline that often concerns itself with the conservation and stability of textual information. At the same time, modern computational methods of analysis clearly demonstrate a need and a use for new forms of representation and interface with text in the digital environment. In this chapter it has been suggested that a recently-developed basic representation of text – the text-as-graph model – strikes a useful middle ground between the computational and the physical representations of text, providing as it does for both conceptual complexity and computational tractability. We demonstrated its use by application to one of the most common needs within textual scholarship: a representation of the variation of a text. Whereas textual variation is usually expressed in print by a dense and intricate apparatus criticus that appears together with other forms of apparatus in accompaniment to a presentation of the canonical text, through the text-as-graph model the text can instead be made tangible through an innovative graphical interface focused specifically on this one aspect. From this work we draw the conclusion that, rather in contrast with the converging practice of digital textual scholarship as a remediation of the codex, the potential of digital interfaces should be explored by developing an array of specific interfaces for specific textual functions, in accordance with views on (digital) textual scholarship as a form of disciplined play (Rockwell 2003). In this way, textual scholarship can embrace the richness and disruptive freedom of the digital medium as a positive force for knowledge, rather than an externally-imposed threat.

Chapter 8

Working at the Intersection of Computational and Scholarly Styles of Thinking: an Autoethnography

> "What do you think machines have to do with your problem, can you elaborate on that?" – ELIZA
>
> (Weizenbaum 1966)

8.1 Introduction

My thinking is rooted in theoretical scholarship as much as it is in software engineering knowledge and experience, and in the memorable intricacies of personal and institutional politics. It took over fifteen years of de facto interdisciplinary academic work to amass the practical engineering experience, academic scholarship experience, and reflection to produce the insights I have tried to capture on the pages of this dissertation. The way this work came about thus differs from a more usual four-year PhD, and this experience has importantly also influenced my research and results. It is for this reason that it is useful to add an autoethnographic chapter to the body of this dissertation. It allows me to retrace how and why my thinking changed over the years, and this will be useful to understand how textual scholarship and software engineering interact and shape academic practice and convention.

I will not audaciously pretend that my personal experience should be a model for how the interaction between academic software engineers and textual scholars should occur. Ultimately, my motivation is this still relevant remark by Christine Borgman: "Why is no one following digital humanities scholars around to understand their practices, in the way that scientists have been studied for the last several decades?" (Borgman 2009). Although work has been done (e.g. Antonijević 2015), much more ethnographic work is needed to understand how software engineering, computer science, textual scholarship, and institutional politics interact. This type of work is pivotal to inform the agenda setting of the interdisciplinary intellectual work that goes on at the intersection of these domains – because developing an intellectual agenda is itself pivotal for this developing interdisciplinary field as Willard McCarty (2014) pointed out. Without it a field might just linger on merely as a methodological niche, being mostly auxiliary to other fields and over time dissolving into them.

Within the large documentary task Borgman called for this chapter will be just one data point, but I want it to be a richly annotated data point, as I think there is as much value of knowledge in experience as there is in analytical and quantitative reasoning. Towards the end I will draw some conclusions about the interaction between academic software engineering, computer science, and textual scholarship and where I think it should take us from where we are. It would be false pretense to suggest that those conclusions are solely and carefully based on scientific reasoning, because they are certainly also informed by subjective personal experience situated in a specific academic context. It is that experience that I try to document as a token scientific record with this autoethnography.

The application of autoethnography as a method commands a short defense, I suppose. The presupposition that any form of autobiographical documentation can result in valid scientific knowledge, has not gone unquestioned. Amanda Coffey, although acknowledging that the situated self of the investigator necessarily influences the fieldwork of any ethnographer, warns that "it remains debatable as to whether utilizing ethnographic strategies to write autobiography really 'counts' as ethnography at all" and that some would argue "that such texts are not 'doing' ethnography at all, but are self indulgent writings published under the guise of social research and ethnography"

(Coffey 1999:155–156). Obviously, I disagree. I would rather appreciate the ethnographic value of any form of information. My academic background may well play into this, as scholars of historical literature usually have no choice but to appreciate any and all historical documents that may have survived time's relentless jaws – happy with every scrap of evidence they can find. More importantly, as Galison (1999) has pointed out, "Objectivity is romantic": the distanced, disinterested, objective academic observation is an invention of the late nineteenth century connected to the rise of mechanical registration (photographs, phonograph, etc.) of data. But as he concludes: "There is no neutral strategy of machine usage followed by an ethical evaluation of it. The machine is moralized from the get-go." And the most moralized machine may very well be the ethnographer's self.

With that in mind, there is really little difference between the utility of autoethnography and that of methods perceived as more reliable – provided that sufficient explanation is given about how the information was gathered so that the information can be put to use responsibly. We should be careful about the reach of our claims. No, one cannot claim this account here as a universal truth, but yes one may use it as additional information to support my argument for a particular scientific agenda in digital textual scholarship.

As for accountability and explanation, Anderson (2006:378) defines five key elements of analytical autoethnography. The first is that the participant-observer must be a complete member of the social world under study. The second is that the autoethnographer should be aware of a deeper level reflectivity that stems from the reciprocal influence between ethnographers and their settings and informants. That is: both as an involved researcher and observer, the autoethnographer has an influence on and is influenced by the context he is studying. The specific kind of analytic reflectivity applied by the autoethnographer, according to Anderson (2006:382) "entails self-conscious introspection guided by a desire to better understand both self and others through examining one's actions and perceptions in reference to and dialogue with those of others." As a third requirement Anderson contends that an autoethnographer should be visible in his ethnography. Ethnographers have traditionally been an invisible but omnipresent narrator in the texts they produce. "Autoethnographers should illustrate analytic insights through recounting their own experiences and thoughts as well as those of others. Furthermore, they should openly discuss changes in their beliefs and relationships over the course of fieldwork" (384). Fourth, for an autoethnography to be analytical it should be rooted not just in self-observation but beyond that in dialogue with those in a similar (or the same) situation. Finally, analytic autoethnography is defined by "its commitment to an analytic agenda" (387). Therefore this autoethnography should not be about me and my experience, self-absorbed or indulgent, but it should concentrate on the inferences that may be drawn from one's personal observational data and what these may yield as wider-reaching contentions concerning the social context one is working in.

So, in what follows, I will hold myself to Anderson's rules of the game.

8.2 An Autoethnography: Intellectual Spoils of Interdisciplinarity

8.2.1 About Trying to Retrace Key Experiences

In practice, research is not the smooth, neutral and well-defined process that many people seem to expect it to be. Many forces shape any particular research project besides the researcher(s), the data, the hypothesis, and method (cf. e.g. Latour 1987). In my experience, the power exerted by the politics inherent in institutions, projects, and funding is considerably larger than many researchers in the contexts I worked in acknowledged or seemed to be aware of. So are the politics and ethics of individuals striving towards personal goals. All of these forces together shaped the interaction that took place on the intersection between software engineering, computer science, and textual scholarship that I witnessed over my fifteen odd years of experience in an academic context. However, only traces of this more holistic perspective found their way to the pages of the previous chapters. Therefore I want to recapture some of what I regard as key experiences that shaped my thinking and reasoning in what went before as much as research and scientific literature.

My initial objective in researching the interaction between software engineering and textual scholarship was to prove a point, but the truth is that the actual point eventually found me. In the course of discovery the investigation evolved my style of thinking. My experience is therefore not just one of writing a dissertation, but of becoming a different type of thinker. I started out as a hard-nosed empiricist, and ended up leaning much more towards philosophy. In my experience there is a sensation in science somewhat similar to the Stendhal syndrome, the name for various real physical effects that seeing art may induce in people. There were a number of moments in which my perspective instantly shifted so radically that the sensation became distinctly physical – as if I could feel neurons click into their appropriate places. They seem mostly to have occurred when someone made – in my view – a key remark in research related conversation. I want to record these moments not just because they changed my thinking, but because I increasingly understood that I was part of the problem.

It took time for the problem to find me. But things also needed to be arranged for me so that I could discover the right tools to approach the problem I was causing. I needed one or two helpful gestures and some time to stumble upon the tool which is called Science and Technology Studies before I could actually start making some sense of my evolution of thinking in some slightly systematic fashion. Science and Technology Studies, or STS for short, consists of the critical examination of science, technology, society and the relations between them. Hence STS was a good fit for the subject I had in mind. But it was also a good fit because more so than the arguably somewhat bookishly oriented scholarship of humanities – especially the field of literary studies wherein I mastered – STS pairs theory with the empirical observation of human (and technology) behavior, in other words the stuff that my experiences seemed to be made of.

Long before I was knowledgeable about STS, however, I started my research by doing and making, by building software for humanists. But the results of this haptic thinking did not work very well. All the innovation and technology I offered textual scholars mostly seemed to generate resistance, however many tools and skills I threw at the problem. I had plenty of tools and skills though, and a good dose of positivistic technological thinking to accompany them. I had code repositories, I had user centric agile development methods, I had test-first programming, I had communication and documentation skills. I even had management backing. But none of this seemed to interest the textual scholars in the least. What remained was frustration and cynicism about all the failing effort, and most of all cynicism toward the lamentable humanists for being resistant to the blessings of modern digital technology and methods. Of course, what I was really lacking – without knowing it – was a proper framework to systematically observe what was happening at a socio-epistemological level, and I had no proper way of reflecting, of systematically trying to understand those observations.

Due to my lack of a systematic reflection at the time I have no formal scientific record of observations from the projects that played at the intersection of research and engineering for the years that passed between roughly 2003 and 2013. Lacking any STS knowledge prior to 2016, the autoethnography I write is necessarily an analytical autoethnography of hindsight and could therefore maybe better be called an analytical autoethnographic reconstruction. In an attempt to mitigate the most pernicious effects of memory distortion and subjectivity I apply the constraints for autoethnography as listed by Anderson (cf. above).

8.2.2 About Sensing but Mislabeling a Real Problem

It took a number of key experiences before I could first see that the problem was not so much one of difficult technology, lack of skills, and resistance to change. These may look like problems at a practical level, but in essence they are symptoms of a more fundamental problem that is socio-technical and foremost techno-epistemological in nature. At the time it was impossible for me to see this because I lacked the language and concepts that are needed to reflect on such issues. I did not know the term "socio-technical", nor the body of critical theory related to it. All that would only come a scant decade later. I thought of software in the usual terms of the industry: users, requirements, functional design, technical design, graphical interface, process, and data. In some sense the social dimension was brought to the table when Willem van den Ende – a very experienced programmer I was working with at the time to develop scholarly software – remarked that "the problem is never technical, it is always social". Admittedly I still simply assumed, like Willem, that this social problem existed wholly on the side of the humanities scholars as users and had to do with resistance to change and especially technological change.

Around this time, which is 2005, Ronald Haentjens Dekker and I were developing various digital tools at the Netherlands Institute for Information Sciences (NIWI). One was called eLaborate. ELaborate started out as an, in hindsight, over-scoped content management system for humanities and social science data (Beaulieu, Van Dalen-Oskam, and Van Zundert 2012). ELaborate was not itself a tool for computational analysis but rather a tool for data entry. Our idea was to support textual scholars by making the creation of web-based scholarly editions easy. At the same time we would ourselves benefit from the side effect that born-digital editions generated machine-readable representations of historical texts that could also be used for computational analysis. From a methodological point of view this approach seemed completely sound. The idea of web-based digitally born editions that textual scholars could create collaboratively were directly inspired by what we knew from literature around Computer Supported Collaborative Work or CSCW (Greif 1988).

I remember that Karina van Dalen-Oskam, who was leading the project, called a progress meeting for stakeholders and key users where we demonstrated an intermediate release and argued computational approaches to the analysis of textual data. A discussion ensued about the interpretative aspects in which we contended that these were certainly not taken away from the researchers. Rather we pitched eLaborate as a tool supporting and not supplanting conventional research, and argued that the best analytical tool was still the researcher. One of the key users – a literary researcher – nodded in great agreement at that and added: "Indeed! And deep long thinking!" More than anything else, the belligerent tone of voice has made sure I can recall this moment at will ever since. It was the first time I was directly confronted with the fact that creating and offering digital tools could actually provoke not just resistance but outright aggression, born from frustration and misunderstanding between developers and users. To me this still stands out as a key moment because it made perfectly clear to me that this was not a problem that would evaporate because of the technical quality of the tools we were making. This type of resistance would require more than just the usual documentation and training.

When Karina and I, and later also Ronald, moved to the Huygens Institute (which then still went under its former name "Constantijn Huygens Institute for Intellectual History and Textual Scholarship") the social science modules of eLaborate were transferred to other partners, but development of the core functionality that was concerned with text transcription and annotation was continued by me and Ronald. While online content creation now seems a rather mundane basic affordance of the Web, in 2005 "Web 2.0" technology was still quite new. Google had only just acquired an upstart that later was to turn into Google Docs (Wikipedia 2020). At that point in time eLaborate was at the leading edge of development with regard to web based authoring and collaboration, especially in the textual scholarship field.

The director of the institute, Henk Wals, perceived digital publishing as a viable and adequate means to supplant the exclusively print-based high-quality scholarly editions and reference works that were produced by the scholars in the institute. Reshaping the scholarly output as web-based publications made sense from a high level managerial perspective. It was expected that it would reduce cost, and that it would do so without lowering quality. Rather the opposite: digital editions would improve the support and affordances for scholarly editing, and with that their quality. At the same time it would allow the Huygens Institute to reposition itself as an institute at the forefront of digitally innovative scholarship.

ELaborate found some enthusiastic users and supported several projects successfully. Most notably a group of volunteer transcribers produced the transcriptions needed for a web publication of an early print late Middle Dutch translation of *De proprietatibus rerum* by Bartholomaeus Anglicus (1485 CE). The edition launched on 5 March 2010 (Werkgroep Middelnederlandse Artesliteratuur n.d.). In hindsight it looks like eLaborate offered value mostly to scholars and editors outside or only semi-related to the Huygens Institute. Indeed for such scholars eLaborate offered a free web based transcription environment with collaborative aspects – i.e. working on the same shared document – that were an improvement above mailing around copies and versions of documents. And possibly in time eLaborate would also offer them a free or at least affordable outlet for the resulting editions. However, the welcome the tool received from textual scholars working in the institute was at best lukewarm.

Ronald and I as developers and Karina and I as scholars did not really grasp why textual scholars in the institute were reluctant to work with eLaborate. Although the tool had its shortcomings we were positive that the benefits – i.e. no version conflicts, a shared working environment, and the ease of working anywhere – should be ample enough to live with the downsides, which would be temporary anyway as development would continue. We mostly still figured that technophobia and resistance to change were the main causes of the less than expected success of the tool. We saw confirmation of that in the fact that we had a few willing users, a large group of skeptical scholars that might be convinced still, and a few scholars – mostly older ones – that outright refused to use eLaborate or any software that progressed beyond email and word processor. To us that seemed like a normal "demographic" make up of a group adopting a new technology.

8.2.3 About Getting it Wrong

What I was not able to see at that time was that from the perspective of most textual scholars – who had never had any experience with computers apart from using a word processor, browsing the internet, and using e-mail – eLaborate meant a complete disruption of their well established workflow. For us putting text directly on the web was a logical next step given our knowledge, use, and experience of Web 2.0 technologies. To editors who had done no digital work prior to that time at all, it looked like a reckless proposition to hand over the stability of print publication and a methodology sophisticated during centuries rather than decades to a technological fad with very doubtful claims to sustainability. More importantly we have never, I think, fully appreciated or considered how reductive the work model of eLaborate appeared in a methodological sense.

Through teaching and practice a scholar develops a professional understanding of the structural and analytical concepts that pertain to the practice of scholarly editing. The scholar learns to think in great detail about such concepts as “document”, “text”, “transcription”, “author”, “context”, “audience”, and about the relations between them. As much through education as through experience a highly personalized systematic praxis develops for finding, selecting, reading, transcribing, annotating, and contextualizing historical sources. Eventually a textual scholar develops an unparalleled individual literacy in observing and describing select historical sources. On a conceptual level there are many commonalities between these individual systems. All editors, for instance, take notes and make transcriptions. But how these notes are taken, how transcriptions are made, and how, for instance, annotations are categorized, is primarily left to the scholarly editor. That is not to say there are no rules and no boundaries. Several well-known standard works introduce the work of the scholarly editor in fascinating detail (e.g. Mathijsen 2003; Greetham 1994). But a recurring motif in these works is a full recognition of the uniqueness of any text worth editing as to materiality, content, provenance, historical contexts of use, and interpretation. No two texts are the same, and therefore scholarly editors can get only so much mileage out of standardized rules for their craft. At some point all scholarly editors are confronted with the heterogeneity of their material. At that point it is the responsibility of the editor to account explicitly for the choices that were made in the editing of the text, and then to proceed conscientiously to a representation that the editor deems adequate.

In hindsight, again, eLaborate failed to acknowledge the idiosyncratic aspects of scholarly editorial work. It also only supported a rudimentary one-dimensional linear notion of text as a series of characters. It lacked a convincing way of delineating the multidimensional structures found in real world documents and texts. It did have very advanced annotation facilities that allowed for arbitrary and overlapping annotations of text. But that did not sufficiently compensate the lack of easily adjustable layout of published texts. In all this meant that eLaborate must have felt to scholarly editors as forcing an oversimplified model of text on them, one that would never allow for the subtleties found in source texts that they felt needed to be captured and expressed. Arguably editors would have been able to overcome many of the shortcomings they felt were part of eLaborate, either by workarounds or by acquiring some literacy in HTML (the markup language in which webpages are written). Attempts at training and teaching however fell short and demand was insufficient to create a critical mass of eLaborate users inside the institute.

One cause that sunk eLaborate was our assumption that all scholarly editorial work could be reduced to one model of text and text processing. Surprisingly, maybe, we had arrived upon a one-dimensional model exactly because we understood the problem of text not being one-dimensional. We were well aware of the post hoc rationale of the Text Encoding Initiative's[1] XML[2] based hierarchical perspective of text by DeRose, Durand, Mylonas and Renear (DeRose et al. 1990). More importantly we were convinced that the TEI's hierarchical model – even if it was a de facto standard in the field of digital scholarly editing – was not a good fit for text that is in essence multidimensional (cf. Buzzetti 2002; Buzzetti and McGann 2006). Text has various layers of structure and meaning that are interconnected – chapter structure versus narrative structure for instance, or literal versus symbolic meaning, or the way syntax contributes to semantics. Because the meaningful elements of such layers and the layers themselves essentially are networked and not neatly nested inside of each other like Russian dolls, it seemed to us that a model that demanded an exact hierarchical structure of text was unfit to capture so many useful structures.[3] That is why we specifically chose to model texts as strings of characters that could be arbitrarily annotated while annotation could be stacked and overlapped in any combination. Note however, that we also did not provide a means of networking annotations – although we did provide a means, at first crude, of categorizing. The upshot of these choices was, I suspect, that we inadvertently created an impression with the textual scholars in the institute that we were working from very simplistic

[1] The Text Encoding Initiative or TEI is the de facto leading standard for marking up digital texts according to scholarly conventions. The TEI Consortium maintains and governs its guidelines. (Cf. https://www.tei-c.org)

[2] XML or eXstensible Markup Language is TEI's primary digital technology to markup text. (See https://en.wikipedia.org/wiki/XML for background and https://www.w3schools.com/xml/xml_whatis.asp for a quick primer.)

[3] We were also acutely aware that it is technically possible to express any other structure than a single hierarchy in XML. The way the TEI and XML are designed however, and especially the rationalization of these by DeRose et al., reveal a clear predilection for a strict hierarchical modeling.

assumptions about the complexity of text.

There are two opposing forces at work here. One is a force directed at abstraction and generalization, which is generated by the engineers and digital humanities researchers in this story. The other force is driving towards concreteness and specificity, and is produced by those who identify most as textual scholars.

A key moment that made me appreciate this opposition occurred when I returned – more or less – to the institute after a three year period where I had been largely uninvolved with the institute's developments because I was managing a large project that, for the most part, was externally oriented. Mariken Teeuwen (a classicist at the institute) was one of the researchers with a more benevolent attitude towards eLaborate. Although she judged that many things were still suboptimal she did use eLaborate as a primary tool during the collaborative editing of Martianus Capella's *De nuptiis Philologiae et Mercurii* (Teeuwen 2008; Teeuwen 2011). The text of "Martianus", as we colloquially got used to calling it, has been preserved in a fascinating document. The space between the lines of the core text bristles with glosses and annotations, while the margins are covered with more elaborate commentaries and Tironian notes (refer to figure 4 in chapter 7 for an example). While talking about the pros and cons of the tool, Mariken explained to me how she had actually used the tool to describe the text of Martianus. This use was quite counter to the intended use, which was to transcribe all the text on the page of a document into a single transcription window and subsequently to highlight the annotations in the original and mark them as an "annotation" (figure 8.1). Instead, Mariken's solution was to "lift" the original annotations from the base text in the window and to put them in full into the windows that were meant to hold annotations by the current editors, and specifically not the annotations of eleventh century scholars on Martianus' text (cf. figure 8.2). Mariken's creative use was born from the very practical consideration that it would have been cumbersome in the scheme originally intended to relate contemporary annotations to the medieval text. It was of course not the case that Mariken and her team did not understand the intended use of the annotation facility. It was simply that the creative use as a more direct description of the original document made a whole lot more sense to them than the envisaged use.

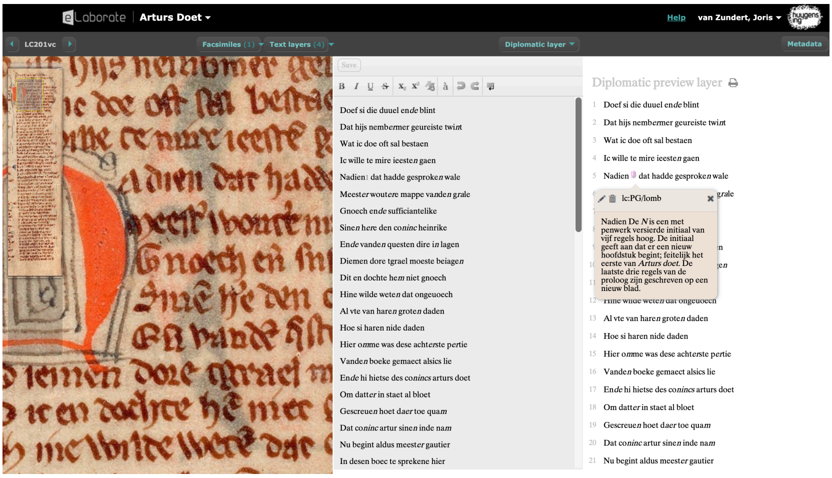


Figure 8.1: The "engineers envisaged use" of eLaborate's annotation function; the annotation box (pop up) is used to harbor an annotation by the scholarly editor.

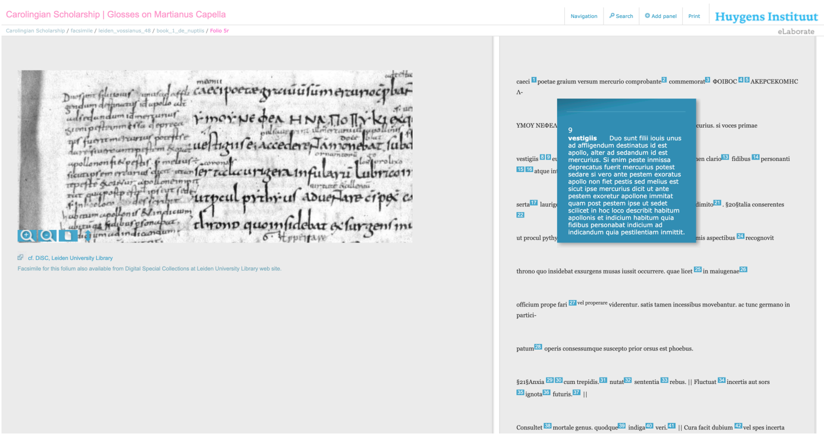


Figure 8.2: The "creative scholarly use" of eLaborate's annotation function: the annotation box (pop up) is used to harbor an annotation that was found in the margin of the transcribed document.

What I realized myself at that moment, if not for the first time then at least for the first time lucidly, was that code creators and textual scholars have opposing interests in the same technology. Engineers are trained to support automation by abstraction. That is: if there is a possibility to support a repetitive task or to capture an often occurring pattern in data they will create a digital function or object that will act as a model for that task or that pattern. Because many unique instantiations of the task or pattern can then be captured by the same digital object this is – from the engineer's perspective – a tremendous scaling and efficiency benefit. Ideally all annotations belong to the class (or category) of "Annotation". In this way the propensity deeply ingrained by training in engineers to draw abstractions from concrete objects in the interest of automation is also, by definition, reductive. Engineers err on the side of broad inclusion. In this sense we, as engineers, had expected that all metatext (i.e. all elements, information, and metadata that pertain to the text but are not actually in the text) would be represented by Annotation Objects. Conversely we had expected that all things in the text would be in a Text Object. This approach to modeling information is exactly opposite to how the textual scholars in the institute were thinking about text. Through their training and teaching they have understood that every text worth editing deserves the best scholarly description possible. In this approach there is no such thing as "abstracting away": all words, all characters, all individual structures of all categories within a text ideally should be explicitly labeled and named. In this approach of course an annotation is still an "object of category Annotation". However, that is not the essence of it, as engineers might reason. For scholars the fact that an annotation is an instance of the "class" annotation is its least important aspect. Far more important are the specifics: where, for which (part of the) text, by whom, what is it saying, when was it made, etc. This is why the eLaborate model applied differently than intended made for a better description of the text that Mariken was working on, because it lend the description more "immediacy" and therefore seemed scholarly more precise. Textual scholars prefer a more specific and more precise description of the specific historical source over some abstraction in service of digital (re)mediation. Their primary concern is not digital mediation but a precise scholarly description of the text. The automation, modeling, or digital support of the workflow is at best a "nice to have".

From a high-level overview and once again with the aid of hindsight it may be easy to gauge that these forces in principle do not have to work against each other but should be perfectly complementary, keeping textual scholarship highly precise and specific, yet allowing it to be digitally mediated. Applied carefully, respectfully, and with mutual understanding, digital tools might have had this effect. At the time we were initially developing eLaborate however, we probably created the impression that we were doing away with almost all specificity in text description. To be entirely clear: we were not. We intended for the specifics to be described precisely in various annotation objects. The textual scholars however preferred to see all specific textual constructs they perceived in texts visualized in the interface as distinct visual objects or representations. That is: they would have liked to see visual representations and digital objects expressing "page", "paragraph", "verse", "annotation", "line", "sentence", "chapter", "speech", "character", etc. etc. Instead, what they got with the first version of eLaborate was "facsimile", "transcription", and "annotation". This created a misunderstanding about the overly simplicity of the model used. We intended the scholars to use the annotation object as a dynamic modeling object so that they themselves might allocate any object, pattern or structure they found important to mark out in the original text and document. That is: we would have them selecting a paragraph, hit the "annotate" button and have them type in "paragraph" plus the reason why it got annotated. The reasons why the model was as generalized as it was are thus clear and proper from an engineering perspective. But the impression from the scholarly perspective, equally clear and proper, was of a rather underwhelming model, almost entirely reductive with regard to the rich multidimensionality of text.

Much of this relates to what is also described in chapter 3. In that chapter another example where the desired specificity clashes with the scaling benefits of abstraction serves to underpin the definition of an analytical notion of paradigmatic regression. What may be useful to infer from these cases is that "creative use" is an important signal that should not be neglected. Rather, as for instance Willard McCarty also argues, the moment a tool breaks down it reveals important information (McCarty 2005:41–43), in this case on the misalignment between models and goals of engineers and scholars.

8.2.4 About a Dialogue That Did Not Happen

Would there have been ways to better appreciate these different perspectives of programmers and scholars? The idea that the OHCO (Ordered Hierarchy of Content Objects) approach was fundamentally flawed was deeply entrenched among us, the technicians – we thought so both as scholars and as developers. Chapters 4 and 7 of this book should also make abundantly clear that I, notwithstanding XML's clear and great utility, remain convinced of this. I remember that we understood that textual scholars wanted more clearly defined text objects and visualization of these in the tool's interface (pages, columns, notes, etc.). We met them halfway by providing curated lists of annotation categories so that any text could be annotated as being a page, a column, etc. But working with these categories was often found tedious and cumbersome. I think we were all stuck at a purely functional level with respect to the tool at hand. The discussion between the scholars and developers always converged on missing functionality, the ease of use, how things looked, the rigid workflow etc. As far as I can recall, a real scholarly-technical discourse on text models never truly emerged in that time.

It is not a given that eLaborate would have been more successful if we had been able to leverage the more fundamental discussion, because it is not at all a given that we would have convinced the textual scholars that the eLaborate model offered more freedom and was thus in principle better for expressing multi-dimensional aspects of text. However, it was certainly not going to happen without such a discussion. And we might have acknowledged that the discussion kept hovering at an insufficient level of reflection earlier than we did, because there was a small group of textual scholars within the institute that started to work on TEI-XML transcriptions at around the same time. Scholars that were hitherto not using digital methods or techniques, or were even until that point actively eschewing them, started to experiment with XML-tools and the TEI text model. This, I think, should have signaled to us that the problem was not resistance to digital tools per se. Because why would they turn to Oxygen (a "word processor" for XML) at all in that case? Entrenched as our assumptions were, we interpreted it as resistance and competing solutions, not as non-verbal moves in a more fundamental discourse on text models.

8.2.5 About What Method Meant

I may be getting a little ahead of my narrative however, because I am quite sure that at that point in time I would not have been able to consider the problems that eLaborate met from a reflectional point of view with regard to text models. One reason for this is that I had not at all appreciated what "method" meant to the textual scholars in the institute. Moreover, I had not quite figured out how to relate the concept of method to the work of software developers or scholars. My understanding of "method" was no more than "the way things are done". As an analytic frame that is not very useful.

Much later Matt Burton (then a recent PhD in STS turned postdoc, now lecturing at Pittsburg University) would relate theory, method, and technique in the clearest way I have ever heard: "Method is theory plus a technique." That is: method is the systematic application of a technique to verify or falsify a theory one has about certain research data. The neat function of this definition is that it ties together three of the most important aspects of research work in a clear and reciprocal relation. And also only much later I would come across a wonderful phrase by Federica Frabetti: "I propose a reconceptualisation of the digital humanities as a field that can and should try to understand the digital in terms other than [...] the technical ones[...]. Such an understanding must be pursued through a close, even intimate, engagement with digitality and with software itself" (Frabetti 2012).

Understanding that an intimate exploration needed to be undertaken and having a clearer picture of the relation between theory (model), method, and technique would have been extremely useful. If so, we might have reflected more adequately on what the differences between eLaborate's model of text and other text models meant. But as it happened, in our software development context "method" meant foremost agile development principles. "Agile" is a set of development principles that tries to maximize customer value and user centric design in order to deliver best fit functionality even in contexts with highly volatile user requirements (Martin 2003). In an academic context new to digital methods user requirements change quickly and often. Therefore we chose "agile" as our development strategy. But "agile" puts the focus very much on function. Users describe requirements in the form of "stories", e.g. : "If I press a button with a red cross the corresponding column is closed." This meant that in every meeting between scholars and developers it was not model that was foregrounded but rather functionality. The point of the methodology should have been to lead us to a discourse on text model, but the specific method was such that it would focus effort elsewhere.

I think that at the time we were developing eLaborate, method – in a research sense – was for us simply equal to digital technology. I was convinced that any tools we would offer textual scholars would mean an improvement simply because they were going to be digital and web-based. Those techniques were going to fast-forward textual scholarship method in the institute to the twenty-first century. Thus better tools would lead to better method, where "better" was somewhat naively defined as "digital" and not, as it ought to be, in terms of some new hermeneutic perspective or a specific epistemological gain. When prodded for exact benefits we kept returning to a "work anywhere, collaborate anywhere, explore anywhere" mantra – and we would add the affordances of the computer as an analytical tool once the texts would be machine-readable. The epistemological gains of computer-supported collaborative work (CSCW) seemed clear to us. Also nobody seemed to deny the uses and gains of being able to find historical resources online. But whether the method used to put historical resources online was also scholarly valid and well argued was a question that we had only a feeble, circular sort of argument for.

However, I am getting ahead of myself again. I keep meaning to return to three key remarks that were made before all of this, before I realized that the rationales of engineers and textual scholars created opposite dynamics. I think that without those remarks I would have been unable to identify the problem as such. Although these remarks will sound simple, almost self-evident, they strongly influenced the reflectivity of my thinking.

8.2.6 About Becoming More Reflective

At the time, in 2009, I was managing a large project called Alfalab, which was a Royal Academy subsidized project to leverage CSCW as a means to strengthen the collaboration between the humanities institutes of the Academy (cf. Van Zundert, Zeldenrust, and Beaulieu 2009). Apart from offering an opportunity to researchers of different institutes to work together, the project was a trial balloon for an envisioned grouping of the humanities institutes by the Royal Academy management. The current Humanities Cluster in Amsterdam is what resulted from these plans and early trials. The carrot that drew the institutes – that traditionally have fiercely guarded their autonomy and independence – was substantial funding from the Academy for digital or computational projects that the institutes would bring to the table. A core working group of six to eight senior researchers and project leaders would run the project. The intention was that the individual projects would use tools and data located within other institutes, thus showing the potential of digital methods and techniques to leverage cross-institutional uses of data and technology. The ideal was that something of a humanities lab would emerge where researchers would be able to choose datasets and tools to process these sets with. Hence the name Alfalab, "alfa-wetenschappen" (transl. "alpha-sciences") being a colloquial designation for the humanities in the Dutch language.

In practice this turned out to be very difficult to achieve. Of the six "demonstrators" (i.e. example applications, prototypes, and pilot tools) only one integrated several resources from different institutes to any substantial degree. It was very difficult to come up with research questions that were both useful or interesting and that crossed institutional borders. But even given such questions it turned out to be even harder to curate and service highly heterogeneous legacy data in such ways that different partners would be able to reuse them. And although the idea of tools and data from a high level might have seemed promising, in practice very few really generalizable tools and generic digital infrastructure could be thought of. Most research questions would require highly specific one-time-use workflows, specialized data curation, and bespoke analytical computational tools. Therefore the project proved a success for the individual partners, but it mostly put in sharp relief how onerous it is to collaborate across institutional and disciplinary boundaries. The project demonstrated that digital infrastructure, data, and tools are no magic facilitators of such collaborations.

Given the aims of the project the group comprised foremost researchers and/or scholarly-informed engineers that were at the forefront of digital methods and techniques within their institutes or even in their fields. Two of the researchers involved were Anne Beaulieu and Smiljana Antonijević. Rather in contrast to the others they were not digitally-inclined humanities researchers, nor engineers, but social scientists. Their affiliation was with the then existing Virtual Knowledge Studio that was essentially the STS centre of the Royal Academy (consult Wouters et al. 2013 for examples of its scientific output). Their task was to observe the project and report on the integration of digital methods in the various research flows. They also helped the group by informing them of knowledge and research on innovation diffusion, CSCW, epistemology, evaluation of science etc. – important reflective knowledge that was expected to support achieving Alfalab's aims.

The project leaders would gather on a semi-weekly basis in one of the offices of the Royal Academy to plan and discuss the progress of the various projects. These meetings also provided ample opportunity to discuss problems in a wider perspective. A recurring trope amongst the digitally-inclined researchers were stories about what was experienced as resistance towards digital methods or technophobia in their various institutes. We often talked about how we might lessen the anxiety of researchers towards digital tools, how better training and teaching might be provided to mitigate the clear lack of knowledge and skills that most researcher showed with regard to computational techniques. We also would sometimes be baffled and mystified about how Luddite some of our otherwise so well educated humanities colleagues seemed to be. We even considered if maybe different but mutually exclusive styles of thinking were to blame for the tremendous obstacles we experienced in introducing humanities researchers to digital and computational technologies: were some conventional humanities researchers seriously incapable of abstract thought and could they only relate to text, for instance, as a linear thing?

In short: our discourse was clad in assumptions that the fault ought to be sought with the scholars. Until Anne, taking part in one of our meetings, iterated a number of other possible causes ending her list with: "...or maybe these tools are just not good enough." A remarkably unremarkable remark at first sight. But to someone caught in an echo chamber it was a very effective means of turning the argument back to oneself, in other words: to make it more reflective. It was simply a question we, or at least I, had never considered. I had always assumed that by default the software we built and offered to scholars of humanities subjects was going to be an improvement. Looking back, I have a hard time understanding exactly why I thought this. Apparently the digital nature of these tools brought them well beyond questioning. And certainly their web-based nature (in most cases) gave them an aureola of reaching beyond anything scholars had seen before. That all of that might do actually nothing at all to improve the scholars' methods, I had simply never considered. I think Anne's remark, as simple as it may sound today, caused me to get into a more reflective stance of thinking: for the first time I started questioning what I was doing, rather than making assumptions about what others were doing for which reason.

8.2.7 About Finding a Method

Later in the project Anne made another remark that has distinctly influenced the way I think and approach problems. We were in the more final stages of preparing a scholarly contribution to an edited volume (i.e. Van Zundert et al. 2012). I was satisfied with how the writing was proceeding, as did most of the (six) authors. But at the point I was for the most part satisfied Anne remarked: "I want to go over the text one more time. There's an imbalance in who's talking about whom, the dynamic is rather forcefully from engineer to researcher, and I don't want this to be another technology push article." I had never before perceived what I was doing as pushing technology on anybody. I simply expected that the clear benefits of the tools I presented would convince any reasonable person that they were better tools than the tools of the trade currently in use. The "push" remark unsettled me a little. The effect of it was profound and, judging now, for the better in that I started to think about how my work was being perceived by the scholars I hoped to convince. Until then I was in a pretty deterministic state of mind: if the tools were genuinely better, they would eventually simply win out against obsolete methods. But what if my tools were not noticeably better, and what if I was making not the slightest positive impression on the scholars I was trying to convince? In short: Anne's remarks made me see that convincing was not a matter of (technical) merit alone. But the takeaway for me in her words was not merely that, nor that we needed to "build support within the scholarship community", nor that we needed "better PR". All true enough, but the salient point for me was seeing the merit of systematic reflective thinking.

To be honest I had interpreted most of what I had seen from STS people until that time as academically glorified Luddism. It seemed to me that their stance was pre-cooked and definitely anti-technology. But because of Anne's remarks I understood theirs was indeed a sincere method. A method to examine power structures, networks of influence, and behavior. From the perspective of the scientists being observed by social scientists, ethnography is meta-research, which in my academic context met with sincere indifference most of the time. When discussed, meta-approaches even met with derision from a good number of colleagues whose judgement I normally find important and insightful. But indeed: what if our tools were just not good enough? I started to think that maybe I stood to gain much from reflectively examining my own motives and actions, from examining how what I did impacted the humanities researchers I asserted to aid.[4] Whether I was part of a solution or a problem I did not yet know, but I had figured out that I was part of an equation that itself needed some critical reflection.

As a result of Anne's remarks my interests shifted in two ways. The first was that I got more interested in the actual work of textual scholars rather than in what my master's education had taught me about literary and documentary

[4] A tangential note on reflectivity in computing. Interestingly there is a technique called reflectivity in computer programming. A computer language can be used to interrogate objects active in the language itself, that is: a programmer can make code that examines the properties from some other objects in that language or code. E.g. a programmer might ask from an object what functions it supports, what variable types it expects for these functions, etc. The technique is not very often used and leads a pretty academic existence, but it does have significant application in program language design, performance testing, code profiling etc. The application of such techniques is often called "meta-programming". Thus the domains of computer science and software engineering do encompass reflective techniques and thinking, but these meta-modes are very seldom applied in my experience. Meta-programming is a highly specialized niche. I cannot help but wonder if the little reflectivity I found in these domains and the level of adequate interaction with scholars are related.

scholarly editing: how they did their work, how they motivated it, what they perceived as their method, etc. Although I still assumed that digital technology, also the technologies I was working on, would improve their methods, I began to be more aware of what could be a problematic technology push indeed. Maybe the textual scholars that we were trying to support did not experience our attempts as supportive at all, but as academic politics to push a poorly motivated technology onto them. The second shift followed from that: I developed a more analytical reflective approach to the problem and now got interested in the ways digital technology would affect and change the ways in which these textual scholars were working.

8.2.8 About Identifying the Real Problem

The thought that just maybe our tools might not be good enough was reinforced a little later as the result of another event. We accompanied the launch of the "demonstrators" that the Alfalab project yielded with a symposium on digital humanities. One of the invited speakers was Annamaria Carusi who has a highly interdisciplinary profile in humanities, philosophy, medicine, and in digital approaches bridging these domains. For us she was therefore an example of successfully working at the intersection of the humanities and digital technology. Her address on digital tools and humanities theory (Carusi 2011) was critical. The main thrust of her argument was that digital tools in science, medicine, and the humanities were doing poorly because computer scientists and software engineers had taken little notice of the humanities aspects that are involved with developing such tools. To underscore her argument she stated: "They have been working on this for more than forty years, and what have they gotten us? Predicate logic! Something we knew already for about two thousand years! Thanks to philosophers – like Aristotle." The statement was arguably intentionally hyperbolic. But it is actually a quite fair depiction of how far computer science has managed to come: the overwhelming majority of general-purpose computer languages (Java, C, Python, Ruby, and all their ancestors and derivatives) are indeed rooted in first-order logic. Only much later I would appreciate that this makes them a poor fit for hermeneutics oriented humanities computation (cf. Gamut 1991:75–76). Nevertheless, Annamaria's contention and Anne's remarks were eyeopeners for me: I was to genuinely question the capabilities of the tools we made.

Key remarks are not made only during research meetings proper. They may crop up in any context. During drinks following a presentation and networking event, for instance. They are not all voiced as constructive dialectic either. Sometimes people voice contentions that seem so utterly flawed that it is hard to decide where to start pointing it all out. Imagine a small circle at a scientific reception. A director of the Royal Academy, a director of one of the Academy's institutes, a renowned scientist in the field of natural language processing, one of computer science, and me. We got caught up in a discussion about the perceived slow take up of digital and computational methods in the humanities. At some point one of the scientists resolutely waved aside the more convoluted point one of the others was trying to make. He argued it was just natural to have some early adopters, a large chunk of people that would only slower adopt the new technology, and a number of laggards that would maybe or maybe not convert much later. "Exactly," said one of the directors, "I firmly believe this resistance is an issue that will pass with time. It's a generational thing." "Indeed," nodded the computer science guy, "the older researchers may well let this treat pass them by, but the next generation will take to it with great ease." Obvious this was just talk at some reception – maybe even in part the wine talking – but there was asserted profundity in the statements by these leaders of fields and institutes. What struck me was the complete and unshakeable believe in the superiority of digital approaches as compared to conventional scholarship. I would only later learn that this is technological determinism at its most positivistic. But it was the generational thing that drew my more immediate attention. To understand why I, at least intuitively, knew they were blatantly wrong about that I have to digress a bit.

8.2.9 About the Myth of the Digital Native

As it happens I have always had a propensity to be intrigued by people *not* getting a computer, or software. I was raised in a family where the computer was already a technology that was well integrated in household live from the mid 1970s onward. My father was a self-taught expert and later professional in the hardware and software that would be used in secondary school teaching. The home were I grew up contained early self-built microcomputers, a PET CBM[5], later C64's[6] and the first IBM personal computers[7]. I played *Pong* before I had ever touched a tennis racket. I knew how to operate a punch card reader before I took my first multiple choice test that applied punch cards to register the answers. When *WarGames*[8] was released I was eleven and I found it kind of cool that I was actually programming computers, although I recognized I was not even close to anything like cracking systems – but I understood the basics of code in a few computer languages (Basic, TurboPascal, and assembly language[9]).

In that time, a youth spent in a household close to several computers, thanks to my father, also meant walking around in a laboratory for the observation of first contact between humans and computers. People would come visit because they knew my father had some of these things that were interesting and apparently useful. Some asked if they could be educated a bit by my dad, others found a convenient place to use word processors while not at work, and there were my father's friends and colleagues who fed their mutual professional hobby. From these observations I can report the two things that computers and software excel at: pissing people off and convincing people they absolutely suck at operating a computer.[10]

Some twenty years and two generations of students later, during the latter days of my university education, nothing had changed. Computers still excelled at pissing people off, they just did it at an exceedingly large scale, and with impressive devotion. Software had dramatically improved its skills to drive people nuts. Ubiquitous peripheral devices, like external drives, modems, and printers added to mystifying (non) behavior of digital

[5] See https://en.wikipedia.org/wiki/Commodore_PET
[6] See https://www.c64-wiki.com/wiki/C64
[7] See https://en.wikipedia.org/wiki/IBM_Personal_Computer
[8] See https://en.wikipedia.org/wiki/WarGames
[9] See https://en.wikipedia.org/wiki/BASIC, https://en.wikipedia.org/wiki/Turbo_Pascal, and https://en.wikipedia.org/wiki/Assembly_language
[10] I do realize this wording may be perceived as unacademic. I suppose "inducing technology related anger and instilling a feeling of utter ineptness" might have done. In the interest of academic precision I chose to keep the original formulation.

technology that was often sure to provoke physical aggression. Because I was one of the geeks that actually had figured out how to tame these weird machines, I was one of those persons that was often called upon to help. I even turned that capacity into a short career at a computer retail firm, which paid for part of my university tuition.

These experiences taught me that the divide between "getting" and "not getting" computers and software is not a divide along the often suspected hard boundary between sciences ("gets computer") versus humanities ("will never get it"). It is also not a divide between young and old. Kids can play games very well, much better than their parents. And they are avid social media users. But they really are not better at all at understanding computers or software (cf. Scott 2013 for more anecdotal evidence). And it is not in any way generational: every generation has the same tiny percentage that naturally takes to how computers operate (or are operated) and the same far larger majority that rather would beat the crap out of the machine most of the time. Of course there are more scientific narratives to back this up (e.g. Kay 1993:81).

Because of this background knowledge and experience I instantly knew that what was said at that reception made no sense. The "problem" was not going to resolve with the influx of a new generation of researchers. Maybe it was the utter positivistic attitude that rang so loudly in these remarks that made me appreciate for the first time an important underlying problem. Deterministic thinking is a form of myopia. Technological positivism locates problems of technology acceptance outside the technology and its creators, creating an illusory positivistic drive through the perceived inherent superiority of the technology. But to understand problems of technology adoption and resistance one needs to accept one's own role as an actor in what is essentially a socio-technical system: you have to understand that you are fully part of the problem. Saying that it is "just a generational thing" is locating the problem well outside yourself and the technology. It is turning a blind eye to what is at least half of a complex interaction. This is what these remarks made me see: if you do not consider yourself as part of the problem as a technology creator, you rob yourself of the ability to mitigate it. Just discarding the problems the humanists had with us and our technology was not going to help us. I had to make those problems exactly my problems.

8.2.10 About Appreciating Differences

It was around this time that we engaged with IBM in exploratory negotiations about a possible cooperation between a computer science department at a university, humanities scholars from the Royal Academy, and engineers and researchers from industry (IBM). Not long before Steve Jobs had more or less said that the humanities were the next big thing (Lehrer 2011), and people were seeking what the implementation of that idea might look like. Sitting at the table at various times during this exploratory phase created an excellent opportunity to witness a gulf of misunderstanding between humanities scholars on one side of the table and computer science and artificial intelligence people on the other side. Computer science people are meticulous, mathematical people. They are interested in the computability of problems and solving the heart of the mathematical problem. While they are very clever and skilled at that, the rationale behind the subtle reasoning with very sparse information that scholars are used to is beyond them. A scholar may produce a well-wrought thirty pages of argument about a single piece of information in a text, tying clues and leads to each other making a case for a certain plausible point of view.[11] In this sense scholars are maybe not so much interested in finding definite solutions to problems, but foremost in creating multiple perspectives and speculative interpretations on "What may

[11] Hans Westgeest's article that links two Middle Dutch documents to the same author because they share a single piece of information found nowhere else (cf. Westgeest 2001:22) still stands out to me as a very good and convincing example of this approach. I take the liberty of adding another tangential note: there is proper cause to systematically designate the way many scholars construct argument as abductive reasoning. Unfortunately in the field the term is not really innate. It describes, however, quite precisely the main angle of attack scholars often use to develop a solid argument where there is hardly any evidentiary material to be found. It consists of using every possible scrap of historical evidence to construct a line of reasoning that is plausible, i.e. not necessarily provable but indeed most likely given the scant knowledge available. Using the term would create a continuum of reasoning spanning the sciences (predominantly inductive, but using other styles too), social sciences (predominantly deductive, but using other styles too), and humanities (predominantly abductive, but using other styles too). Creating a continuous understanding of styles of thinking overarching the scientific domains (cf. Kwa 2011; Crombie 1995) would mitigate the uninformed, dangerous, and damaging two cultures divide (Snow 1998[1959]) that has brought so much unproductive divisive thinking to science and society.

have happened here?" Their method is not looking for universal truths at all. Most accept that truth and fact are situated and historicized social constructs: what counts as truth is determined by Fortuna's favorites, and history is not deterministically moving in a direction of progress.

At the time I would not have been able to describe the situation as in the above because I lacked the analytical vocabulary to relate what was happening to notions such as "analytical", "rationale", "universal truth", "situated". But what I clearly sensed during the IBM episode was that the scholars and the computer science people around the table talked distinctly differently about problems. The hard-nosed scientists would address problems as things that could be divided into smaller problems that individually would have satisfiable and provable solutions. Putting together the solutions of the smaller problems necessarily would lead to either solutions of grander problems or to strategies to attack these problems. In other words: they were always solution-oriented, which seems to be a general trait of most anything computer related (cf. Morozov 2013). The scholars however, would tend to discuss what-if scenarios. They would probe and examine possible views and angles on problems, almost like wine connoisseurs figuring out if the problems were actually palatable. This would lead to a typical unsatisfactory sort of conversation. The humanists would table a specific problem or category of problems in their field and the computer science people would immediately start throwing solution-oriented strategies at it to see how solutions could be made evidentiary and empirical – maybe to the indignation of the humanities scholars who were much more interested in how different arguments with the problem as the topic could be constructed as an intellectual exercise to interpret and understand the humanistic aspect, dilemmas, moral grounds, and ethical considerations of the case in question. They were not interested in the solution-to-the-problem but in the problem-as-a-problem: in its ontological meaning and its epistemological potential; in how it might advance our different understandings of the problem.

I have Charles van den Heuvel (historian of science at the Huygens Institute for the History of the Netherlands), and Sally Wyatt (Professor of Digital Cultures at Maastricht University, who was at that time also affiliated with the Virtual Knowledge Studio) to thank for bringing to my attention some of the essential differences that I had not grasped until then. Charles at some point talked about "ephemeral and heterogeneous data", which sounded esoteric to me. Data for me fit in categorial, discrete, or continuous variables: measures of size, distance, density, time, place, person, and so forth. It took me a while before I understood that humanities researchers did not think at all about data in only such constrained computable terms. Humanities data are both more particular and less constrained. They are more particular in that they are situated, i.e. located in place and time, bound to historical context, and possibly – actually likely – subjective. They are less constrained in that sometimes the precision of a particular data point may be less relevant, so that in some cases "ca. 409", "5th century", and "13 October 409" are equally valid.[12] It is therefore fair to say that humanists mostly do not work with data-as-data, but that they combine, examine, and interpret information, which may be defined as data-plus-context (Thaller 2018). Where physicists might be inclined to subdivide measurements into as accurate as possible times, locations, persons, temperatures etc., the humanist is often interested in a very particular heterogeneous combination of data where sheer availability may be more decisive than exactitude. Measurement of basic data itself in the humanities is tricky and difficult because it is an interpretative process, often historical data is a best guess based on information written down in a different context centuries before. But for the humanist these are even "merely" preparatory moves, after which follows the more important move of interpreting the data as a complex of information.

Without understanding this difference humanities reasoning may be taken by more hard-nosed scientists as a form of improper or imprecise empiricism, whereas it is really a matter of two fundamentally different understandings of what data and information are. Confusing one for the other will lead to misunderstanding each other's methods and aims in research. Computer scientists and engineers discarding this difference may fail to see how methods cannot simply be exported from the computer science domain into a humanities domain. Trying to do so will fail with frustration on both sides. Human-

[12] Which is computing wise a bit of a conundrum because it is unclear and highly context dependent when a measure becomes positively invalid. Does "ca. 409" mean that 400 is still included? And 550? Humanities computing therefore cannot be as straightforward as simple arithmetic.

ists are not primarily interested in stringent solutions to specific problems, and they are not especially looking for patterns that allow them to predict future outcomes. That approach is decidedly hard-sciences empiricist. Johanna Drucker (2011) has justifiably pointed out that importing such methods unamended into the humanities would mean supplanting a refined humanistic method with a naive scientism.

Sally Wyatt succinctly captured the above when in one of the IBM exploratory meetings she said: "Humanists do not solve problems, they create perspectives." That thought would later become the recurring motif in the white paper that would eventually result from the involvement with IBM and other partners (Millen et al. 2013). It would also be an eyeopener for me.

8.2.11 About Taking Textual Scholarship Seriously

Slowly thus, remark by remark, experience by experience, my perspective shifted. I began to appreciate that paradigm shifts, in the way Kuhn (Kuhn 2012[1962]) described them, and deterministic diffusion of innovation do not happen just because we want them to happen. Looking back I have also no reason to believe the scholarly editors felt anything like an epistemological crisis. But they surely felt threatened. If not by management then certainly by us, engineers, who must have come across as the leading edge of a technology that was pushed upon textual scholarship. The discussion on the side of the self proclaimed innovators was fueled by terms like "digital", "computational", "revolution", "augment", "improve", "resistance", "support vector machine", "software", "good practices", "programming", "critical mass", "new generation". Almost none of it must have signaled that we were also interested in a discourse, in a dialectic between digital and non-digital textual scholarship. We also failed to adopt sufficiently the concepts and terminology that the textual scholars used, which might have created enough trust and cooperation for what we were trying to do. If a productive epistemological trading zone had been established we should have noticed a far greater influx of scholarly terminology into our vocabulary rather than the other way round (cf. Galison 2010:39). Most of all we did not grasp sufficiently the essential difference between a scientistic approach and a pluriform perspective view on text and textual scholarship. To solve this the dynamic had to change – as Anne Beaulieu had indicated – from technological push to mutually constructive dialectic. And both we and the textual scholars would have to become far more reflective about our own motivations, aims, methods, and techniques.

A final nudge that changed my attitude from programmatic innovator to analytical thinker was given by Paul Wouters whom Douwe Zeldenrust and I had asked about supervising PhD work that we intended to undertake based on our experiences in the Alfalab project. Paul, unsure of what I intended to work on, suggested I write a "would be" introduction to my dissertation to create some ground for mutual understanding. After reading my draft Paul smiled and put it plain and simple: "Content wise that seems fine to me. But your angle should be a bit more cogitative. A little less programmatic, a little more analytic." Sometimes eyeopeners come in really helpful obvious shapes. Basically Paul told me to try to understand more and contend less. I still had to increase my analytical stance a few notches.

So since that time, I sought to understand. Textual scholarship foremost. I returned to some basic readings in textual scholarship (e.g. Mathijsen 2003; Greetham 1994), reread works that are on the intersection of digital methods and textual scholarship (McGann 2001; Buzzetti 2002; McCarty 2005; Landow 2006, etc.). I talked to researchers in the field that did interdisciplinary work. Susan Schreibman, for instance, Fotis Jannidis, Barbara Bordalejo, Peter Robinson, Tara Andrews, Dino Buzzetti, and many more). These conversations were important to understand how those scholars understood the interaction between textual scholarship and the domain of computing. And often they also served to dot my i-s and cross my t-s. I remember, for instance, a conversation between me and Susan Schreibman over a lunch in Amsterdam where she kindly prompted me to keep a clear distinction between textual and literary criticism.

This renewed immersion in textual scholarship made it clear to me that methods of textual scholarship can only be understood as reciprocal. If the aim is to establish a text philologically or to analyze it in a literary sense, understanding and interpretation of the text are a necessity. But interpretation is by definition something that involves an interpreter, a reader. And because the interpreter, the text, and its creator are all situated, interpretation cannot be other than intersubjective. This period also made clear to me that if I was to understand textual scholarship on any fundamental level, to be able to understand how digital techniques and methods might be useful in any fundamental way for textual scholarship, I needed to understand interpretation on a more fundamental level.

8.2.12 About Interpretation

I think it is important to note that I discussed my work on this course towards understanding textual scholarship often with my friends and colleagues in the then software engineering department of the Huygens Institute, mostly with Ronald (the same as mentioned before, who by that time had become a lead developer) and with developer and project manager Gertjan Filarski. In their projects, like me, they had experienced mostly what appeared to them as resistance from textual scholars in the institute. The tone of our conversations would be quite sarcastic, to the point of cynicism. Nevertheless they, and sometimes other software developing colleagues, due to their role and experience were formidable sparring partners in reflecting on the interaction between developers and scholars. Gertjan was later to become head of the department of digital infrastructure when several humanities institutes clustered in Amsterdam after a few more years. During the preceding years he grew into an outspoken opponent of anything to do with textual scholarship that skewed towards a pluriform or relativistic view on method or subject. We have had several disputes about what he perceived as an obsolete method – which is to say, conventional textual scholarship in general – and what I had started to regard as a different style of scientific thinking. Surprisingly then maybe, I have to thank him for turning me back to the topic of hermeneutics which I had left behind me a long time ago, around about 1996 when I had to peruse it in more general terms for my master's education in Dutch literature and linguistics. I had found it a topic or method that was hard to understand and, to be fair, a bit esoteric.

To understand interpretation one cannot evade the topic of hermeneutics, which is the theory of text interpretation and which historically goes back towards ancient text interpretation and biblical exegeses. There is a clear hermeneutic tradition from Aristotle's *Poetica* via Augustinian interpretation of religious texts, Schleiermacher and Dilthey's Romanticism, Wimsatt and Beardsley's rejection of authorial intent, postmodern philosophers like Derrida and Foucault, semioticians like Eco, and so on. Understanding this tradition as a philosophical foundation of textual scholarship became the topic of two chapters of this dissertation (i.e. chapter 2 "Screwmeneutics and Hermenumericals: the Computationality of Hermeneutics" and chapter 6 "Author, Editor, Engineer – Code & the Rewriting of Authorship in Scholarly Editing"). Tracing this tradition became a reflective argument in itself to accept the subjective nature of all interpretation. Stephen Ramsay (2011c) argues on hermeneutics that humanists always have been in the business of constructing plausible histories from subjectively-selected facts, from information much contended by authorities, educated opinions, and scant evidence. I also rooted my understanding of the tradition of hermeneutics in the traces of histories written by yet others. On comparison my interpretation seems not incommensurable with the interpretations of others, but it is still my interpretation and understanding. Or, with Korzybski, it is not the territory but a map. Thus tracing the tradition of hermeneutics resulted in new understanding about a multitude of possible perspectives on heterogeneous and non-neutral text, the situated pluriform meaning of signs, the subjectivity of any form of interpretation, the endless semiosis of semantics. Seeing how all of that connected to the methods of scholarly editing increasingly made it harder for me to view such editing as a scientific approach that could easily be turned into a uniform process.

8.2.13 About a Brave New Model

And yet, integrating through unifying is exactly what we set out to do in the institute time after time. We tried a unifying approach with eLaborate. Then we tried again with Alfalab. And after that, newly-appointed director Lex Heerma van Voss and I set out once again to harmonize the different approaches to editing that were applied by the textual scholars in the institute. Henk Wals moved position shortly after the deliverables of the Alfalab project had been launched, becoming director of the International Institute for Social History (IISG). He left behind a legacy in the form of a new institute merged from the Huygens Institute and the Institute for History of the Netherlands (ING). In manifest obviousness the new institute had been named The Huygens Institute for the History of the Netherlands (Huygens ING). Less obvious was how the various methods of scholarly editing applied in both institutes could or should be harmonized. From a managerial perspective the merger seemed sensible. Both institutes have a strong tradition in editing and publishing historical sources. Both also maintain smaller but equally excellent analytical strands of research. The managerial challenge that Henk faced was to enlarge the analytical line of work while maintaining quality and output of the editorial strands of work.

Henk Wals has always been a strong advocate for digital methods and techniques as a way of scaling humanities research, of having new research questions emerge, and to be able to answer humanities research questions that until now seemed infeasible to address given the amount of data needed or the sheer complexity of data. As recounted above, eLaborate was to be a pivotal tool in scaling the production of scholarly editions, both by ensuring scholarly editions would be henceforth digitally born (reducing the need for the painfully slow production process of very expensive print publications and retro-digitization) and by converging the various methods of scholarly editing used in the institute towards one harmonized method "recognizable as a Huygens ING method" (Wals et al. 2012:23–24, transl. from "een herkenbare Huygens ING-aanpak").

After some ten years, continuous substantial personnel effort, and large financial investments – easily summing to over half a million Euro, though probably much more – several sites in the eLaborate platform testified to the various projects undertaken to produce digital scholarly editions with added content from analytical research on the historical texts of those editions (cf. e.g. "Published" n.d.). However, when Henk Wals moved and Lex Heerma van Voss became director, we had not succeeded in scaling the scholarly editing process. Tremendous effort had been put in digitally remediating the work of scholarly editors and textual scholars, but essentially without changes to the scholarly method. The results were scholarly editions digitally remediated to look as much as possible as their print counterparts (cf. also chapters 3 and 4) – and most of the times editors would only grudgingly accept the non-print result. In sum this meant that the remediated editions required – in *addition* to the expertise and effort of textual scholars – the knowledge and effort of software engineers, web developers, and graphic designers to translate the scholarly work to the Web – which arguably rendered them as expensive if not more expensive than print publications. Really scaling this process to achieve a shorter turnaround and a greater efficiency for edited materials still required a new model for the process of editing.

When Henk left the institute Lex Heerma van Voss and I took this as our challenge. Henk had already come up with a model for added scientific value. He presented it as a pyramid (see figure 8.3), which likely was inspired by Ackoff's "data-to-wisdom" pyramid (Ackoff 1989). The base of the pyramid was constituted by the physical documents from heritage institutions like libraries, archives, museums, etc. that cared for them in their collections. The scholarly editors of Huygens ING became involved in the layer above that. They created scientific resources based on the documents available through the services of the layer of "GLAM" institutions (i.e. galleries, libraries, archives, and museums). It was there where the impact of eLaborate was expected to be most valuable. Finally, the top of the pyramid consisted of analytical and synthesis oriented research by the institute's leading scholars on historical topics of science, politics, literature, culture, and so forth.

Over the course of various discussions Karina, Lex, and I adapted the model by adding and refining layers, creating a rather box-like model in which layers were stacked and where each layer added its own specific scientific value. After a popular Indonesian dessert – a kind of densely layered cake – Lex called it the "spekkoek" model. The idea was that fully-enriched digital scholarly resources would emerge at the top of a stack of process layers. Each layer is connected to a specific data processing or scholarly task that builds on the results of the underlying layer.

In the case of scholarly editions one can imagine the physical archives and libraries as a first layer. Digitization is a second layer. Transcription a third.

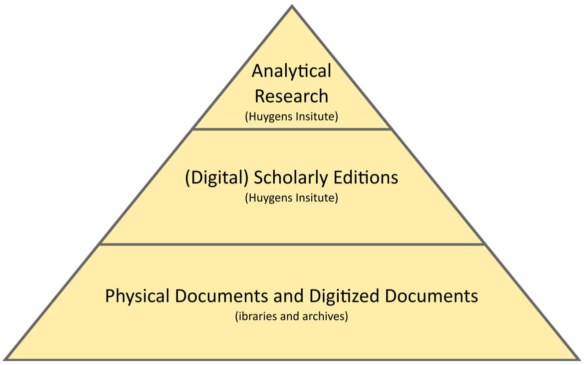


Figure 8.3: Pyramid of added scholarly value, after an idea by Henk Wals.

The key is that every layer is "thin" in that it is connected to one specific and definable task. This decouples the layers from each other so that to execute its task on a resource a layer only needs the finalized input of another layer, while it is not dependent on any information or tasks in another layer for any of its internal processing. Decoupling ensures that the process of an individual layer can be evolved to a maximum efficiency.

As a whole the model strongly resembles the Unix pipe architecture, popular in the IT world, where any process or transformation on data is independent of any other (Ritchie 1984:1581–1591). All processes function as a black box with input and output specifications only. To minimize and if possible eradicate any dependencies between layers the task in each layer needs to be as specific as possible. The model as a whole is strongly process-oriented rather than product-oriented. Thus, for instance, digitization is primarily seen as a generic serviceable task, and not so much as a specific action in the creation of a unique scholarly edition.

In this model the very first layer – atop any of the ones where we presume the physical collection work to be situated – only takes care of primary digitization, meaning that print and manuscript material are digitally photographed or scanned at high resolution and are stored in a repository with minimally sufficient technical metadata and minimally-sufficient bibliographic metadata. Technical metadata will usually already be embedded in an image file.

Although a file format like TIFF also allows for limited but in most cases sufficient inclusion of arbitrary metadata as well, bibliographic metadata will usually be stored through some other digital structure (database, document store, XML files, and so forth) for more convenient handling. Some stable URI identification scheme will allow unambiguous identification of an image and its metadata. Creating these objects is a clearly defined task with scholarly value, requiring a number of high-end technical and bibliographic skills that can be executed as an independent task (cf. e.g. Hughes 2004, in particular part 2, chapter 8). On top of this first digitization layer, a layer for primary transcription is stacked. The only input this layer requires is the output of the previous layer: a digital image and its bibliographic metadata. It is the specialized task of this next layer to create initial transcriptions of the digitized images. To do so OCR (Optical Character Recognition) can be applied, which will work for modern sources to a high degree of precision but will usually produce worse and worse results the older the historical resource is that an image is derived from.[13] The aim of this initial transcription layer is not to provide a transcription that is completely trustworthy from a scholarly perspective, but to provide quick access to the textual and pictorial content of the digital images. These initial transcriptions have no authoritative scholarly status of any kind. The rationale for this layer is that not all research needs high-end quality transcriptions, and that often even poor OCR is enough for text mining systems to create some high-level overview of what is available in a particular set of resources, or to isolate documents that might be connected to particular topics. The idea here is thus to preempt the very long turnaround of creating the highest quality scholarly transcriptions and editions that require much more time for meticulous and authorized scholarly editing. While not denying the value of such editing the initial transcription layer provides early open access to initial – likely imprecise but for some scholarly tasks sufficient – digital transcriptions of document images. Although mechanized transcriptions might not carry a scholarly authoritative stamp of quality, they do have scholarly value that is added in isolation in this particular layer.

[13] It should be noted that the technology of HTR (Handwritten Text Recognition) as applied, for instance, in Transkribus (https://transkribus.eu/Transkribus/) and crowdsourcing solutions as applied, for instance, in Vele Handen (https://velehanden.nl/) are working miracles to mitigate these limitations.

The general approach will be clear by now: the "spekkoek" model adds layer upon layer, where each layer is associated with a specific scholarly task that adds scholarly value and quality to the historical resource. On top of the raw transcription layer one can imagine a layer in the stack where rough and mechanized transcriptions would be improved by human transcribers. These might be experts knowledgeable about specific script in the case of ancient or medieval documents, in other cases transcribing could be crowd-sourced to volunteers. ELaborate was envisioned to play a major role in this layer. The layer above that would be where predominantly the scholarly editors of Huygens ING would be involved. They would correct transcriptions based on their expert paleographic, historical-linguistic and subject knowledge. In a next layer they would add annotations. And so forth. In theory the topmost layer, through a kind of bubbling-up process, should see the emergence of high-end scholarly-validated authoritative editions of historical sources.

8.2.14 About Another Failed Dialogue

Implementing this model as a solid and shared methodology in the Huygens Institute never succeeded. But it was in fact never a set aim. Sketching out the model was foremost a first attempt at making explicit, comparable, and discussable the large number of "idiolectic" methods that were rooted in formal scholarly training as much as in individual experience and tacit knowledge. We carefully refrained from initiating a discussion with a premeditated opinion that the methods in use were invalid with regard to digital media or that they were in any scholarly understanding flawed. The intention was to seek consensus for a model by inviting scholarly editors to an institute-wide discussion. Preliminary work for the discussion was to be done by a working group that would generate an inventory of editing methods in use in the institute. From this common ground a suggestion would arise about how to amend the existing methods with a digital counterpart. The goal was to find true consensus on an inclusive digitally-remediated method. Lex assigned me the task of tabling the topic of a shared digital method and I intended it to be a true remediation (Bolter and Grusin 2000:60; Karlsson and Malm 2004:13). That is: not just a medium shift but also a renegotiation of the method with regard to the differences between the media involved. Lex would probably not put it in STS terms like this, but I am convinced that his sincere intention was also to work towards consensus. Even though we realized that some darlings of methodology might be hurt in the process, we were convinced that enough advantages and advances in the digital method would make the collateral damage bearable, even worthwhile for all scholars and scholarly editors involved.

The discussion went nowhere however. The discussion got desperately stuck in endless reiterations of the same arguments. And it appeared to be impossible to find even the simplest thing in various methodologies that looked sufficiently similar to have the scholars agree that it could be defined as a generalizable task that might be digitally remediated. In all, seven preparatory meetings were organized. But failing to find some common thread and being deadlocked in repeated argument, the initiative petered out and was eventually abandoned.

Several external reasons might be put forward to explain why this happened. No formal resources have ever been appointed to the initiative, for instance, although a methodological discussion was bolted down even in the official research program of the Huygens ING (Wals et al. 2012:23–26). There was no budget, time-wise nor funding-wise, that allowed participants to account hours spent on work for the discussion group. Participation was voluntary and time and effort spent would be silently covered by departments' budgets. It was expected that the intellectual exercise and obvious relevance of the methodological discussion for the future praxis of digital textual scholarship at the Huygens ING would be sufficient incentive for senior scholars to participate. And in fact initial participation indeed involved a group of highly visible and experienced researchers of which a large contingent also thought digital methods were something to be seriously considered and examined as a means of practice for the future work at the institute. But the informality of it all ensured that any intellectual effort for the working group became at best an afterthought for scholars that were without exception continuously being overburdened by deadlines on overly long to-do lists and being flooded by urgent interventions in whichever project they were working on.

8.2.15 About Incommensurabilities

However, even if lack of formal project space and support might have accelerated the demise of this initiative, I do not believe these were major causes. More important were intrinsic methodological incommensurabilities. It was clear from the repetitious discussion that even though all textual scholars were working on texts and documents, each and everyone's methodology only coincided on a very abstract level. For instance, Selection of sources is an important step for every scholarly editor. But it matters what the selection pertains too. It matters if you are involved with the edition of twentieth century novels or with medieval scholarship.

Marita Mathijsen explains that determining who actually wrote a particular text obviously influences whether it will become part of the editors selection or not (Mathijsen 2003; also cf. Greetham 1994 again) . But this can play out dramatically differently on a practical level. In the case of a highly successful twentieth century dead Dutch white male novelist like, for instance, Willem Fredrik Hermans, the "enormously rich personal archive" and "fascinating [...] correspondence between [the author] and his publishers, his fellow writers, his literary friends, and his enemies" provide a plethora of information about the authorial process (cf. Kegel 2016). This creates its own problems of bias and subjectivity, but it at least allows for some argument grounded in abundant paratext on what to select for a scholarly edition and what to exclude. In other, usually more historical cases even determining who wrote what is difficult. Various anonymously written manuscripts (so called "witnesses") containing a particular text must be compared to establish what should be selected and edited (Mathijsen 2003:123–124). In such cases textual scholarship may become as much bibliographic archaeology as editing the text (cf. Westgeest 2001 for a particular interesting case).

Thus what looks commensurate on an abstract level, i.e. selection, means vastly different scholarly processes on a practical level. More importantly the historical case violates the assumption about neatly separable scholarly tasks. The genealogical provenance work requires highly skilled and specialized knowledge of both the historical text and genealogical methods – consider stemmatology for instance – which is only present in a handful of experts. But in the layer model the resulting knowledge (cf. Westgeest 2001 again) also has consequences for metadata in other layers, e.g. the layer of collection formation and the layer of digital imaging. This does not necessarily invalidate the model itself, but does show its limited encapsulation of the actual praxis and dynamics of textual scholarship..

Thus even if two scholars agree that what both do at some point in the editorial process can be called, for instance, "transcription", the actual practice of "transcribing" may be two rather incommensurable activities: the one being executed in some archive with pen and paper next to a box full of scraps of documents, the other being carried out behind two computer screens, one with a word processor open and the other depicting a digitized microfilm. The very nature and properties of the documents and the texts that they contain differentiate what "transcription" means. For one scholar it means knowing Middle Dutch grammar, dialectic variation, and lexicography by heart, and being an expert paleographer of medieval Dutch manuscripts. For another it means developing intimate knowledge of the handwriting in the diaries and letters of a modern author, and amassing sufficient contemporary contextual knowledge on topics that the author wrote about, to be able in the end to translate idiolect phrasings written in a peculiar, hermetic, personal stenographic system.

8.2.16 About a Digital Tool Breaking Down

A series of discussion meetings can hardly be called a key moment. Key process seems more suitable as a label. This process in any case made me experience something I had never witnessed: failure of abstraction.

Abstraction is one of software engineering's most powerful tools. It is a method of modeling by which many particulars can be molded from an abstracted super class. Imagine, for instance, one is developing a drawing program that can create circles, rectangles, and triangles. One can construct three distinct objects, each of these having lines of code to position the specific shape, to draw it, scale, fill, store, delete it, and so forth. But many of these actions are not specific for either shape. This is most easy to see for e.g. positioning, deleting, and storing: the information whether we are talking about a circle, a rectangle, or a triangle is irrelevant to the position of the centre point, the deleting of the object's digital information, or the storage thereof on a hard disk. From an engineering point of view it is therefore convenient and efficient to create a super class, e.g. one called "Shape", that has functions with specific lines of code for positioning, deleting, and storing. All specific shapes will then be created as "offspring" objects from this super class, and they will inherit the functions for deletion, positioning etc. Only particulars unique to the specific shape then need to be expressed as lines of code in the object associated with the specific shape. Thus Shape-Triangle will have different code in a function called "draw" than Shape-Rectangle, but they will share the exact same lines of code for the function "delete". One can dispute endlessly whether abstraction is fundamentally an inductive or deductive approach. Most likely an inductive or deductive propensity is dependent on individual choices by programmer. When coding myself I tend to think of my theorizing about the world as deduction and about my practice of coding as induction.

The salient point of this with regard to the methodological discussion on textual scholarship was that this method failed utterly when trying to create abstractions for the activities of textual scholars. It is rather straightforward to define commonly-understood abstract tasks in textual scholarship: selection of sources, reading sources, transcription, annotation, and so forth. However, once one delves below that level one is immediately confronted with a garden of forking paths that is different for every combination of scholar and source, as the example of transcription above illustrated.

Failure of a tool is an excellent opportunity for learning (McCarty 2005:41–43). This failure of abstraction and the process of experiencing it up close in these meetings was key in understanding how hermeneutic the practice of textual scholars actually is. It is hermeneutics to its deepest core, it finest veins, its smallest act. If hermeneutics was turtles, it would be turtles all the way down. The scholar that develops an intimate knowledge of the handwriting in the diaries and letters of a modern author, does so not just to be able to read the handwriting for transcription. Through the very experience of learning to read and then reading the manuscript the scholar develops a sense for the author's style of writing, and for the content, topics, events recounted in the text, and so forth. The point of which is not just to be able to create the transcription. The reading experience also create affordances for later annotation of the text and for analytical reasoning on the author's developing thoughts.

8.2.17 About the Integrated Nature of Textual Scholarship

The toughest argument against supporting hermeneutics digitally is thus hermeneutics itself: it requires intimate engagement with a text or document in all possible aspects. It requires experiencing its very materiality, the feel for how the sign on the page was formed, how words were strung together. It requires an abundance of contextual knowledge, annotating without end, reading and rereading. Only this experience results in a foundational knowledge of all the aspects of a text from which a viable process of interpretation can start. The back and forth between conceptually charting the meaning of a text on a high level and the minute inspection of a comma that could change that meaning: this is the hermeneutic method.

It is exactly this intimate engagement with all aspects of a text as a whole that one forgoes if one "topples" this process and tries to pull it apart into a series of different tasks, each embedded in some isolated technical layer attributed with a number of dedicated skills. The decoupling of scholarly tasks that would be required to realize the digital scaling of the academic work would also eradicate the hermeneutic aspect of that work. With hermeneutics you cannot have your cake and eat it. In managerial terms the hermeneutic process is an integrated vertical process. It requires from a task owner a specific skillset that is dictated by a specific resource. The process generates new knowledge about a text by progressing through each layer of the "spekkoek" model. This process is not straightforward progressive but iterates many times between layers (tasks). Eventually the process yields a sufficient reservoir of knowledge to produce a scholarly-viable interpretation of a text.

The task-oriented horizontal model assumes that scholarly knowledge can be added independently in each layer. But I found that this is only true to a very limited extent. The first layer of digitization requires technical skills, bibliographic knowledge, and possibly scholarly knowledge on book history and paleography. This might be done in an isolated layer. Also the next level, OCR'ing the text, might still be done in isolation. However, progressing one more level up towards transcription, the correcting of the OCR'd pages requires expert scholarly knowledge of the language, script, and probably also content of the text. This kind of scholarly expertise is not widely available, and it is rather likely that the paleographic knowledge needed in the primary digitization layer for a specific document is tacit knowledge of a person that will also be needed for the transcription in the third layer. Moving upward through the layered model increasingly more knowledge of the underlying layers is needed to create valid interpretations of the text at hand. An argument might be made that once a corrected text is available – a so-called diplomatic transcript – annotation could be done independently of the creation of the diplomatic text. However, in practice the creation of the diplomatic transcript is part of a process that generates a large amount of knowledge about what could and should be annotated to ensure a sufficient understanding of the text. Vice versa the questions that result from trying to transcribe the text drives contextualizing research that overlaps considerably with the annotation task. Moreover, even if the creation of a so called critical text might be completely decoupled from the basic transcription, the work in this layer is still strongly coupled to a higher-level interpretation of the text, because the annotations are the evidentiary material that a textual scholar gathers to produce a scholarly-viable interpretation of the text.

Thus hermeneutics and the praxis of textual scholarship are deeply intertwined. That is: every scholarly task related to a certain document and a certain text is a contribution to the practical part of hermeneutics, which is interpretation. This praxis – and thus hermeneutics – is very much embodied in textual scholars with years of practice and with actual work on a specific historical resource. The uniqueness and heterogeneity of historical documentary and textual sources combines with a uniquely-experienced body enacting hermeneutics that together result in a unique method applied to edit a unique text. Although that method can be understood in the abstract and general terms of scholarly tasks, these tasks are not neatly progressive in time nor is their actualization identical each time when the tasks are enacted. Rather they are imbricated, and they iterate and adapt continuously until such time as an edition can be said to be "finished".

This intertwined and highly resource-attuned nature of hermeneutics is, I claim, the direct cause of the futility of our attempt to remodel the editing process as a stack of unproblematically unrelated layers associated with individual scholarly tasks. For a long time we sought the cause with the textual scholars – perceiving their attitude as resistance to change. The discussion meetings were in part a result of the Huygens Institute merging with the Institute for the History of the Netherlands. The Huygens Institute had a strong history in text criticism and literary editing, while the Institute for the History of the Netherlands had its roots in a tradition of documentary editing. The discussion group was therefore also aimed at harmonizing these different approaches. It appeared from the ongoing discussions however, that the scholars were dead set on pointing out the incommensurability of the two, as it seemed that either species of textual scholar was only able to talk about the differences and specificity of their scholarly practices. The merger of the institutes thus caused the discussion group to be about a threefold methodological merger: the merger of two different approaches to scholarly editing and the merger of those with a digital approach to editing. For a long time this reinforced my thinking that the complexity of the process and in part a resistance from the scholars to digital techniques in general were to blame for the lack of progress in the discussion and the impossibility to harmonize methods.

From a higher-level managerial perspective it is indeed hard to see where the differences are between documentary editing and literary editing or textual criticism. All editors themselves agree that there are similar tasks in all scholarly editing processes. How then can it be so hard to harmonize these task a little? The differences are apparent in the individual praxis of textual scholars, but they are hard to spot for those who seek harmonization and efficiency on an organizational level where only the labels of the abstract tasks are visible. Thus what we failed to see is that each scholarly editing project below the level of abstract tasks indeed explodes in a melee of specific subtasks that are highly attuned to the material at hand.

8.2.18 About the Real Problem

Suppose however that things had been different. Suppose that the discussion group at the level of merging two distinct traditions of scholarly editing would have found complete agreement and harmony – supposedly because the traditions were somehow indeed commensurable. Would we in that case have been able to successfully digitally remediate this methodology according to a model of layered and independent tasks? Again with the power of hindsight: no, we would not have been able to do so.

Recall that management had two goals in mind when it suggested to create a "digital edition machine". The first was scaling up the production of scholarly editions, or at the very least minimizing the turnaround for (digitized) scholarly resources becoming available for further study by researchers other than the editor. At the same time it wanted scholarly editions to become more accessible. "Accessible" in this case could mean two things. Either editions from the institute would need to be more affordable than the forbiddingly expensive print editions – that were so expensive due to the scholarly effort needed, which combined with the low number of copies printed and sold, as they were often targeting small numbers of specialists. Or it would mean making editions digitally accessible in open access. Digital scholarly editions were seen as technology that would actually serve both these purposes: affordable and accessible editions.

Digital technology is excellent at scaling. Once a suitable model has been found for the data or a process it is a formidable tool to reproduce the modeled task or product in staggering quantities with dazzling speed. It can do so where either data or process are uniform or generalizable enough so that it can repeat identical operations on similar data under the same constraints. The trouble with textual scholarship is that this never happens, neither at the level of the word, nor at the level of the text. The hermeneutics of textual scholarship is deeply intertwined with the heterogeneity of its data. Determining what some word in a text means is dependent on the language of the word, the knowledge of the linguistic context of that word, the cultural context of the text, and often also on what is on the page: could these written or typed glyphs actually signal another word, maybe? Reading, transcription, and interpretation are thus interdependent tasks. If one was nevertheless to isolate each task in a layer, assuming it would therefore somehow become scalable, you would find that the task would need a different combination of language, reading, and cultural expertise each time. In such a process there is nothing to be abstracted, nothing to be generalized. Would one nevertheless take to abstraction, then one would eradicate the hermeneutic aspect of the process, or in other words the point of the scholarship involved (i.e. interpretation). A real risk, had we pushed forward, would have been that the constraints of the chosen technology would have become leading. Because the technology can only scale a uniform process based on uniform patterns in data, the actual scholarly process or the scholarly data gets adapted in such ways that the technology will be able to scale it. This is a particularly pernicious corollary of the promise of scale that is characteristic of software engineering – the technology does not scale the actual process or data of scholarship, but an abstracted form that is tractable for the technology.

The conclusion must be that current software engineering approaches and current general-purpose computer languages are not adequate to scale scholarly editing without losing some of its hermeneutic essence. What is possible is supporting scholarly editing in a computer-supported collaborative work fashion, which is by and large as far as eLaborate got. It facilitates a rather convenient way to produce digital-born scholarly editions by shortcutting the tedious task of marking up a digital transcription in the form of TEI-XML while keeping the ability to annotate text. But eLaborate does not scale the core intellectual and skilled process of scholarly editing. It merely enforces an abstracting away from the more precise hermeneutic tool that TEI-XML might be. By doing so it might shorten the turnaround of a digital scholarly edition becoming available online, though I think this has not been convincingly shown in practice. But even if it did, it did so mostly by enforcing a particular uniformity of look and feel of the editions. A uniform typography that lessened the ability of scholarly editors to put the look and feel of their editions in service of a scholarly grounded interpretation of the text (cf. Andrews and Van Zundert 2018).

Many, myself included, have claimed that the key to battling this problem is in increasing the digital literacy of textual scholars (cf. e.g. Ramsay 2011a, Smith 2012, Van Zundert 2018). Chapters 5 and 6 of this book also speak to this idea. That is: teach scholars to code so that GUI based solutions like eLaborate that necessarily enforce a certain uniform model of scholarly editing and edition onto the scholar are no longer needed. Instead textual scholars, empowered to wield general-purpose computer languages, would in that case code and publish their own digital scholarly editions without the intervention of GUI based tools. And although I remain a strong advocate for the idea that all humanities scholars should at least acquire fluency in one general-purpose computer language, I do not think this would fundamentally solve the problem. Because, as Annamaria Carusi pointed out so hyperbolically, current general-purpose computer languages are almost without exception so-called first-order languages. Although it is rather crudely put and not entirely correct, it is also not an exaggeration to say that such languages only support boolean (binary) reasoning. At the very least their make-up invites a developer to tackle any problem as a boolean operation, even if it is quite possible to implement more subtle forms of reasoning with general-purpose computer languages.

A rhetoric of scale and speed is prevalent in mainstream software engineering which is driven by a plain vanilla type of application of first-order computer language in combination with a logic of process automation. This is a type of software engineering that is rooted in a limited binary apprehension of the capabilities of first-order logic for data modeling and analysis, where processes and data typically are modeled after the question of whether stringent, i.e. boolean, comparisons can be made. The basic rhetoric structure of this type of coding is "if X equals Y then Z will be done", where "equals" is defined in absolute terms: either the condition is fully met or completely violated, e.g. "if year is larger than 1066 then label 'post Norman Conquest' ". This discrete logic breaks down on hermeneutic requirements. An inference such as "if genre is Bildungsroman then label 'literary' " is not supported because humanistic categories are rarely absolute but rather more often have unclear, intersubjective, and overlapping boundaries.

It is the boolean nature of general-purpose computer languages that causes uninspired software to be rather demanding with respect to the uniformity of data and process if it is to scale data transformation or analysis, because in its most basic form boolean logic and process automation do not deal well with imprecision, uncertainty, heterogeneity of data, and ambiguity of information. The uniforming nature of current general-purpose computer languages proliferates through much of the reasoning supported or expressed by it, simply because it is de facto the easiest way to program reasoning in these languages, and it is much harder to do anything else. Arguably the majority of industrially-produced software falls within this category because mass production favors uniform processes and products. Such industry software is meanwhile a dominant shaping power in society (cf. e.g. Berry 2014) and might well be seen as a pernicious form of Foucault's disciplining powers (Foucault 1995[1975]).

There are actually ways to circumvent an uninspired application of computer languages, even with the currently most-used general-purpose programming languages (such as Java, Python, Ruby, C#, JavaScript, etc.). There are, for instance, mathematical approaches to deal with uncertainty in the form of probabilistic models (e.g. Bayesian models and networks). These models, well known in the field of natural language processing (Cohen 2016), could well be applied to more hermeneutic reasoning regarding historical textual resources. Knowledge at the fronts and fringes of mathematics and logic bare promise of forms of logic that are better equipped to deal with incomplete, imprecise, and ambiguous information. Modal logic (Pratt 1980; Mastop 2011), possible worlds models (Fagin et al. 1995), and grey information theory (Lin, Chen, and Liu 2004) have all forwarded possible solutions to deal with data and information in ways that relate far closer to hermeneutics than first-order logic proper. All of these, however, remain in the realm of speculative research and none have produced generally available usable software libraries yet. But these advances do show that current programming paradigms need not be the be all and end all of the application of computation in textual scholarship, as has been noted by some scholars (Thaller 2018; Van Zundert 2018).

As for eLaborate and the layered model… We were trying to model and scale the scholarly process of applying hermeneutics, but we did so by computationally providing a digital mimesis of a few rather mundane scholarly tasks and concepts – such as text, transcription, and annotation. By doing so we were enacting a mere feat of cargo cult: we mimicked a few externally visual traces of the hermeneutic work of which the essence is rather a cognitive process in the bodies of textual scholars. Given the severe technical limitations of our computational tools, we would not even come close to representing or modeling human interpretation. One could object that this was never the goal, and that realizing CSCW was indeed achieved. That is a fallacy: the very intention was to scale scholarly textual editing. We – all those involved – never truly realized that this implied scaling the cognitive process of hermeneutics. Attaining that implied objective was futile from the start, given the sticks and rocks we have for tools.

If we are indeed to strive for a computational hermeneutics that will require a far more intimate and deeply involved engagement with quite fundamental forms of computing, logic, and mathematics, both on the part of computer science mutatis mutandis software engineering and on the part of textual scholarship. It will require a sincere shift of minds within these two interacting domains, and the courage to do so mutually and respectfully. Engineers should dare to admit that current software tools – and even the computer languages in which they are written(!) – are severely limited with regard to modeling and reasoning about complex intersubjective information, and subsequently that true computational hermeneutics requires vastly improved computational tools. Textual scholars on the other hand need to start seeing text as an interesting and imprecise form of computational data that yet should be able to be modeled as digital and mathematical information, and that may be edited in different ways than just as a remediation of print text on a digital screen (cf. again Thaller 2018; Van Zundert 2018, as well as chapter 4 of this dissertation). In all of this eLaborate was a failed attempt, doomed from its very beginnings. But failed attempts are part of a path to achievement. Regarding the multidimensional modeling of text the work by e.g. Ronald Dekker, Elli Bleeker, Bram Buitendijk, Astrid Kulsdom, and David Birnbaum (Haentjens Dekker et al. 2018) as well as others are meaningful steps on this journey. But there is a long way ahead of us. Even more ambitious computer science research will still be needed to render hermeneutics somewhat computable.

8.3 Conclusion

I came into textual scholarship with a rather hard-nosed "can do" mentality, reasoning that re-establishing textual scholarship on a digital and computational footing should not be much of a bother. This I found after so many years of experience with textual scholarship: the scientific value of textual scholarship is in its hermeneutic analysis of texts and this hermeneutics cannot be adequately expressed or enacted by the current digital technology we call code.

Three key remarks, I feel, remain to be noted. Not because they contributed that much to my findings per se, but because they provided independent confirmation of those findings coming from non-suspect sources. The first is a remark made by Carl Posy when I attended a dinner he organized at his house for the people that had lectured at a workshop at the Hebrew University of Jerusalem. (Consult Posy 2013 for an example of his work on the philosophy of mathematics.) We were discussing various topics surrounding digital humanities and textual analysis, and at some point Carl remarked: "It is not so much about computation as it is about computability." This remark signified for me that there is good reason to relate the domain of scholarship to computation. In the same sense that Anne Beaulieu wanted to examine the rather unquestioned dynamic between software engineering and computer science on the one hand and humanities on the other, Carl Posy invited me to consider that maybe the humanities hold rather more interesting challenges for computer science than just as an application ground for what computer science can do today. The questions that scholarship can put to computer science are hard and much more interesting than scholars tend to think: they highlight the still limited abilities to reason computationally about humanistic information.

The second one is related to reading Willard McCarty's "Humanities Computing", where he argues that the moment we learn something is the moment our digital technology fails and errors out (McCarty 2005:41–43). This wisdom goes back to Heidegger obviously (cf. Heidegger 2010:334–348 [1953]) and was not essentially new to me. However, it did put me in a state of mind where I started to consider computer languages as a failure instead of as a solution, certainly with respect to textual scholarship. And this proved a more fruitful and productive line of thought than simply regarding computation, computer languages, and digital data as an answer to anything.

The third one is the most recent and came in the form of a blog post by Manfred Thaller (2018). Thaller lucidly describes a number of problems that render current mainstream computer languages inept for any hermeneutic purpose, and that the challenge is to come up with a solution for this.

These three remarks combine into what I find a formidable challenge. If it can be argued, as I did here, that current mainstream computer languages do not support hermeneutics because they lack a hermeneutic nature, then why do we allow that to result in a push on textual scholarship to comply with the uniform nature of first-order logic? Would it not be far more interesting and challenging, in a fully interdisciplinary understanding, to devise an analytical tool in the form of a computer language that is actually hermeneutic in nature from the bottom up? (Cf. also Van Zundert 2018.) It is not impossible, and it is certainly less boring than transforming the analysis of the human textual record into something uniformly bland.

A few loose ends remain to be tied up. What does this all mean, and how analytical is this chapter? To start with the latter: is this writing just some opinionated essay, or is it genuine research? It is analytical in the sense that I kept to Anderson's principles for analytical autoethnography (Anderson 2006) as outlined above. I am a complete member of the social world being studied. If anything this autoethnography is a testament to the deeper-level reflectivity that Anderson points to: I recorded to the best of my abilities the influence I had on methodologcial development in textual scholarship in my academic context and the reciprocal effects of that work on my thinking. I think that qualifies as "self-conscious introspection guided by a desire to better understand both self and others through examining one's actions and perceptions in reference to and dialogue with those of others." Anderson's third requirement is that an autoethnographer should be visible in his ethnography, and this requirement is why at times this chapter takes on a colloquial style. Having a colloquial style seems to be a stock criticism of peer reviewers of many of my articles anyway. I agree: sometimes I do use a colloquial tone. But it is not because I do not know how to wield the so-called distanced and disinterested academic style. My colloquialism is on purpose, especially in this autoethnography: I do not want to give the illusion that my argument is more widely applicable than it is by applying a misplaced academic generalizing style of writing. In this case all I have as evidence is my personal experience and notes – subjective material by definition, but nevertheless valuable to learn from. I have tried to give voice to the insights of others, by focusing on key remarks, and I clearly demonstrated how my beliefs and insights changed through the perspectives and words of others. I maintained an analytical stance by supporting my narrative of experience with ample references from the literature. I also kept an analytical attitude by taking distance from this material in time, by returning to it and rewriting it several times over a period of two years. I witnessed how all versions became indeed less self-absorbed, less indulgent, more honest, and more interested in seeking some truth. The text stopped being about where I wanted to go at some point in time. Instead it became to reflect my genuine interest in where my topic came from and how I got to think about it the way I do now.

The main conclusion I finally draw from all this is that technological determinism blinds and that curing that blindness is pivotal if we are to see the real methodological challenge that lies at the intersection of textual scholarship and computer science. In the case I describe there was a strong belief in innovators in academia that some technology was methodologically better than the methods and techniques that existed. This believe was conveniently in accord with certain managerial needs. I bracket the managerial needs as they are not that interesting in my opinion and largely irrelevant to the main conclusion of my argument. It is this particular blindness of highly-trained, skilled, and clever professional researchers that intrigues me. We had not much more going for the innovations we argued than that the technology was new and digital. It promised scale and speed, but we had no evidence in any particular academic use case to back that promise up. This attitude could be called mild technological myopia still. In the end, confronted with a stubborn hermeneutic methodological tradition, both textual scholars and engineers learned many valuable lessons I think, even if it meant not achieving certain hoped for innovations. This mild myopia is not something I am all too worried about. It creates enough friction of its own in pipedream projects to make forced interdisciplinary projects to implode, be it sometimes at staggering economical and human labor costs. The blindness I truly worry about lies in an unwillingness to investigate respectfully and with sincere interest another person's ideas and perspective. It was only in eventually digging myself a way into "the textual scholar's mind" that I, as an engineer, could start to see the exact properties of a mismatch in methodology. Seeing that, I could acknowledge the invalidity and pretense of an easy process-oriented digital reshaping of scholarly editing. And it opened up an intellectual space that allowed me to recognize the – in my opinion – far more interesting challenge that textual scholarship can table for computer science: developing a formal logic and language that fits hermeneutics rather than shoehorning hermeneutics into current first-order logic languages and thereby losing it all.

One last concern I want to address. I have meanwhile moved my perspective so far to the philosophical side of the spectrum that I am at risk of blaming the engineers and computer scientists for said blindness. But I should stress that I found scholars at least as blind, albeit in a slightly different way. There is anxiety among scholars that they are not being taken seriously by software engineers. Many scholars feel intimidated by shiny and hip information technology. But this intimidation, aggravated by a false modesty about the value of some twenty centuries of scholarship methodology, is inhibiting in detrimental ways. There is a shared task here and no easy sitting back. Textual scholars should raise their voices to ensure that the process of softwarization of text technology progresses responsibly from the perspective of scholarship. In recounting my experiences, it sounds to me like the voice of the textual scholars is heard but that it is somehow also not very present. In chapter 6 of this dissertation I point to the shared accountability and responsibility that both scholars and technology innovators have in methodological innovation. However, it appears to me that often the scholars are kept out of the wind in ethnographic work in DH. Antonijević' work, for instance, at places reads rather apologetically to me where it pertains to the role of scholars in methodological innovation (Antonijević 2015). But if engineers indeed have done little to open their minds to the core principles that underpin hermeneutics, so have the scholars – certainly in my case. They have been utterly unable to voice their methodology and concerns in ways that convey the relevant properties to engineers comprehensibly – often because the underlying assumption is that engineers will not or will not want to understand. Textual scholars should realize that their aims and work bear formidable challenge for computer science engineers, but that it is also their responsibility to develop and expose these ideas in ways that make the challenge clear. Putting that responsibility in the hands of engineers alone is capitulating to a technological deterministic mindset that will render textual scholarship irrelevant when more and more text becomes part of a digital medium. The proper form of digital hermeneutics requires scholars to care about computational literacy and to invest in learning what computational literacy really might do for them. The best way forward is a dialogue, not a monologue as some suggest (Bordalejo 2018; Robinson 2013c).

Chapter 9

Conclusion

The overarching claim I make with this dissertation is that there is a knowledge space between software engineering and textual scholarship that is undertheorized, academically underdeveloped, and intellectually undervalued. The following summary of the most important conclusions of the prior chapters will serve to substantiate this claim. The primary subject of this work as a whole is the interaction between software engineering and textual scholarship, as explained in the introduction (chapter 1). The second chapter, delving into hermeneutics, provided a theoretical background, while chapters three and four are essentially case studies of this interaction, pertaining to the creation of digital scholarly editions and the creation of related tools in the context of the Huygens Institute. The following chapters reflect on this and similar work and are more concerned with questioning academic authority, critique, and evaluation in the context of software development in textual scholarship. The research underpinning this dissertation is strongly interdisciplinary work and it will probably be clear by now that the main argument and conclusions of each chapter do not fall neatly inside this or that disciplinary category. For clarity however, I have tried to gather the conclusions of the various chapters under the headings of three disciplinary categories or perspectives.

9.1 The Textual Scholarship Perspective

Read as a theoretical and practical study of the problems that exist with the currently prevalent way of expressing scholarly digital editions in textual scholarship, my work provides a rationale for a practical solution. Chapter 3 and 4 theorize from experience the limitations of markup models. These chapters add to the intellectual work that arose in the context of "the problem of overlap" in markup languages. On the practical level that problem may seem trivial to software engineers and digital scholars – and they might interpret it as a mere conflation of text structure and the structure of XML. However, the application of XML as a form of markup for text structure in a purely hierarchical fashion, as it was adopted by the TEI based on a post hoc rationale by DeRose et al. (1990), caused a lively intellectual debate about text, structure, meaning, and digital affordances for their expression (e.g. Buzzetti 2002; DeRose 2004; McGann 2004 etc.). On the basis of this literature and my own work I conclude that hierarchical markup is an inadequate fit to capture and express the multidimensional nature of text. In chapters 5 and 7 I propose graph models as a technical solution that exceeds any hierarchical markup model in expressiveness. These chapters also report on some of the practical aspects of my work that contributed to the development of the graph model as a more versatile descriptive model for form, structure, meaning, and interpretation of text and documents. It is exceedingly gratifying to see that the hard and experimental work by me and others with regard to graph technologies is slowly finding a toehold in textual scholarship, as indicated by various publications (e.g. Haentjens Dekker and Birnbaum 2017; Andrews et al. 2018; Neill and Kuczera 2019; Efer 2017) and conferences such as the annual Graphentechnologien/Graph Technologies conferences since 2018, initiated by the Akademie der Wissenschaften Mainz, and the 2019 Workshop on Scholarly Digital Editions, Graph Data-Models and Semantic Web Technologies at the Université de Lausanne).

9.2 Science and Technology Studies Perspective

This dissertation is also the result of Science and Technology Studies labor. The STS perspective calls attention foremost to the coding work that is related to graph models and especially the tools producing the graphs, such as CollateX (chapter 5 and others) and Stemmaweb (chapter 7). The software engineering work involved with these graph-oriented approaches provides a more precise example of how actual software code impacts textual scholarship. A paramount difference exists between the "xmlificaton" and digitization work in textual scholarship and such codework approaches. I found (cf. chapters 3 and 4) that the use of markup results only in a shallow form of remediation which accepts digital technology predominantly as a way to reproduce a digital metaphor of the book on a computer screen. However, the algorithmic work involved with CollateX and Stemmaweb signals a partial shift of agency and control related to scholarly editing. Part of textual scholarship's agency and authority shifts, in more covert than overt ways, first from scholarly editor towards engineer and second, most interestingly I think, from engineer to code (see for this chapter 5, but foremost the section "The author and the engineer" in chapter 6). This shift of agency and control is not as absolute as the title of Lev Manovich's work *Software Takes Command* might suggest. Manovich (2013), like Berry (2014), Coyne (1995), Morozov (2013), and others take rather grand perspectives, arguing how software and digital information may affect societies as a whole. This sometimes truly suggests images of software as an omnipresent "invisible hand". My STS work examines the shaping power of software code at a decidedly smaller scale, and finds that code does not truly take over. Rather, some components of scholarship are trading places. As tedious and error-prone tasks, their control and the associated agency are delegated from the scholar, via the software engineer, to software code.

It should be stressed that this process of delegation progresses in a rather unobserved manner. I witnessed that there is little explicit methodological reflection built into either the ad hoc processes or the formal software engineering methodology that governs projects where software engineers and textual scholars collaborate. Slightly covertly therefore, and unintentionally, this process of delegation passes over a number of important theoretical, methodological and epistemological questions. A number of these matters were investigated in chapters 5 and 6. The main conclusion I draw from this work is that digital textual scholarship and the associated coding work is in dire need of a framework for evaluation. Although it has earned me several remarks that Critical Code Studies, Software Studies, and platforms such as Cultural Analytics[1] long since provide this kind of evaluation, I stand firmly by my

[1] https://culturalanalytics.org/

contention: code peer review and academic acknowledgement of code work are tremendously underdeveloped in textual scholarship and, by extension, in the humanities. None of these domains and platforms provide precise, code level, peer review and evaluation. Their engagement with the scholarly work is at a level of method that is detached from actual software code and considers only the analytic results obtained through code. Whether that code functioned correctly and in accordance with theory and hypotheses is never evaluated.

The conclusion that the evaluation of coding in textual scholarship is inadequate is important, as it pertains to pivotal concerns about who produces knowledge and how we evaluate that. Currently my case studies in the context of the Huygens Institute, and partly beyond, suggest that most textual scholars conveniently assume that authority and control over scholarly work still resides firmly with the scholars – and literature suggests this holds for a wider scope than just the Huygens Institute (cf. Robinson 2016; Bordalejo 2018). But as I argue in chapter 6, scholarship expressed in code results in the same appropriation of revisionary authorship as scholarly editing does. Software engineers and code both produce knowledge, and as such they must not be denied a proper form of evaluation and crediting in any academic system. Both scholars and software engineers should take full responsibility to find and establish ways that acknowledge the scholarly contribution that code and programmers make. A formal means of acknowledgment must allow software engineers – no matter whether they are scholars, software engineers, or both – to be credited and, moreover, accountable for their work.

Another effect that my STS approach to the problem of interaction between software engineering and scholarship uncovered is the process of paradigmatic regression: affordances of unknown or new methodology may become hidden when they are narrowly used to express concepts and terminology of a prior and more familiar paradigm. The digital interfaces, especially graphical interfaces, that stand between computational model and scholar are strange beasts that play a crucial role in this process. They open up as much as they hide the computational model (cf. chapter 3 and chapter 7). At the same time that they make it useable, they distort the computational model in ways that provide endless opportunities for misinterpretation, misunderstanding, and uninformed or plain incorrect use of computational tools. Because they are made by humans they are far from neutral, but they still carry something of a halo of perfect impartiality because this is, however incorrectly in itself, associated with digital computing. Computational models are not neutral and their makers are not perfectly disinterested. Interfaces are loaded and situated. I have found little to no evidence that collaboration between textual scholars and software engineers specifically on graphical interfaces results in any significant methodological gain or diffusion in either direction (cf. chapter 3 and especially chapter 4). If anything, graphical interfaces act more like fences that keep epistemologies well apart and inhibit methodological knowledge exchange more than they enable it. I think this “interface effect” (Galloway 2012) has been especially pernicious in the case of the scholarly digital edition. The fully representational paradigm that digital textual scholarship writ large has preferred for digital scholarly editing has been reaffirmed consistently by skeuomorphic visualization – that is, making the graphical interface look as closely as possible to a physical book, if possible down to mimicking cover, paper, and ink. Arguably the benefits of the tremendous increase in scale of access to (digitized) sources and knowledge do warrant the equally tremendous economical, technical, and personnel resources that representational digital scholarly editions require, even if institutions and scholars still struggle to establish sensible strategies for the sustainability of scholarly digital objects and infrastructure. However, the strong regression towards the representational paradigm – powered and reaffirmed continuously by graphical interfaces – has all but eradicated the appreciation for actual methodological affordances that might be found in a genuine interaction between computing and textual scholarship. To many computing and textual scholarship may seem like irreconcilable foes. Based on my findings however, my contention is that they are the very two components that together underpin a paramount task of methodological scholarship for the next decade: the establishment of a true computational hermeneutics.

9.3 The Humanities Computing Perspective

Obviously this dissertation also results from interdisciplinary work in humanities computing. However, increasingly I have difficulty with the term "interdisciplinary". Like "interface", it implies a separation as much as it suggests some cross-discipline dynamic. I no longer have a use for this separation. In fact, I find it unhelpful and bothersome. The neat disciplinary boundaries that we ourselves maintain place constraints and limits on the use and application of our collective but distributed knowledge. I am a professionally trained programmer, an academically trained textual scholar, and a novice science and technology studies researcher. It is my own hybrid nature rather than the interdisciplinary nature of my work that allowed me to assemble the theoretical knowledge (chapters 2, 4, and 6), practical work (chapters 3 and 5), and experience (chapter 8) that resulted in an understanding of the interaction between software engineering and textual scholarship, informed by multiple perspectives. This in turn allows me forcefully to conclude that there is indeed a terrifying gap between software engineering and textual scholarship. This is the intellectual space I aimed at in the first paragraph of this conclusion – an intellectual space that is undertheorized, underdeveloped, and undervalued. It is the knowledge space of computational hermeneutics.

Mostly by way of chapters 2 and 8 I have argued in more detail how central hermeneutics is to textual scholarship. An understanding and practice of computational hermeneutics becomes urgent in a society and culture increasingly producing its textual legacy through digital text and its digital-native relative: code. It is therefore both surprising and disappointing how little interest there is either in computer science, software engineering, or textual scholarship for the computability of hermeneutics. Scholars have called attention to digital hermeneutics (e.g. Capurro 2010; Frabetti 2012; Meister 1995; Thaller 2018), but a rigorous theoretical and applied program of scientific investigation involving the three related disciplines – computer science (including software engineering), textual scholarship, and science and technology studies – remains wanting.

I have found four main factors that inhibit the various disciplines from progressing productively into this space. The first is the fully representational paradigm and associated technologies of markup that digital textual scholarship has championed as its prime philosophy. A computational hermeneutics for textual scholarship at the very least requires scholars to appreciate the differences in nature and behavior of digital text, its relation to code, and especially the precise performative nature of code with its dual guise of text and executable. However, textual scholarship appears singularly interested in bookish screen essentialism. The fully representational paradigm gives rise to the second main factor: an exaggerated attention to (XML) encoding of texts in what little digital humanities education there is in textual scholarship. Although ever more digital humanities minors and masters have sprung into existence, attention to code and coding is – especially in the Dutch situation – negligible. But code literacy is an essential skill for the next generation of textual scholars (cf. Vee 2013).

Third, computer science (and especially its practical counterpart, industry level software engineering) and textual scholarship sustain a rhetoric that causes a myopic understanding of computing and its uses for textual scholarship. Scholars predominantly sustain a rhetoric of reductiveness about computational methods. While it is true that machine learning, natural language processing, and stylometrics, for instance, are currently rather reductive techniques, this does not mean at all that code must ever be reductive. However, merely pointing this out continually (Johanna Drucker is a well known proponent of this type of rhetoric) is only of limitedly help. Technologists aggravate this situation by emphasizing a rhetoric of speed and scale, by mystifying code (which is actually a rather straightforward semiotics to describe objects and actions), and by sustaining a lack of interest in the potential hermeneutic forms of coding. Attention for possible hermeneutic forms of computer code in scholarship is equally minimal, with the odd notable exception, such as Manfred Thaller (2018).

Lastly, institutional knee-jerk reactions hardly help. It is still too difficult to amass academic credit for digital objects and coding work. This problem is annoyingly persistent. Added to this is an institutional preference for outdated organizational philosophies and structures. Nowhere do I see this dividing force more up close than in the Huygens Institute, now part of the Amsterdam Humanities Cluster. Continuous reorganizations have reaffirmed the boundaries between scholarship and research on the one hand and software engineering on the other. The organizational charts of the successive Constantijn Huygens Institute, Huygens Institute, Huygens Institute for the History of the Netherlands, and the Humanities Cluster have consistently reaffirmed through old-fashioned organizational boundaries who is allowed to produce – or rather who will be credited with having produced – scientific knowledge, and who is not. These boundaries have consistently separated scientific programmers from their scholarly colleagues. Rather than being embedded in research departments they have been increasingly put at an ever-increasing distance from researchers. The separation resulting from such a professionalization of IT services also results in token reaffirmation of these boundaries through ticket systems, administrative resource management requirements, product managers, and so forth. Rather than providing for interdisciplinary work and hybridization of knowledge and skills, this organizational inertia obviously does little to contribute to methodological innovation.

However, I want to close this dissertation on a more upbeat tone. I repeat that I am a professionally trained programmer, an academically trained textual scholar, and a novice science and technology studies researcher. Through my studies and experiences at the Huygens Institute and beyond, over more than fifteen years, I have learned an enormous amount about the interaction between software engineers and textual scholars. The same context, problematic as it sometimes may be, challenges me to keep learning every day, which is a blessing that I am tremendously thankful for. STS method has taught me how to reflect analytically on this ongoing work. These reflections have enabled me to venture beyond my initial personal preoccupations, to look harder, and to question more earnestly. It has been the multidisciplinary context that brought the contours of a "computational hermeneutic void" to my attention. My hope is that this dissertation contributes to an acknowledgement of the intellectual space that exists between computer science and textual scholarship. My expectation is that one day textual scholarship, computer science, and software engineering will truly meet each other in this space. My wish is to explore that space and to understand this beast called "computational hermeneutics" while walking with it.

–Joris van Zundert
Utrecht, 9 September 2019

Summary

Hermeneutics, or the theory and method of interpretation, has a long history that is deeply intertwined with the philosophical and exegetic roots of the type of textual scholarship that developed in Europe. Postmodern relativism and the transient, shapeshifting nature of digital technology are disconcerting forces for a textual scholarship in this tradition that is concerned with human culture memory function and that is therefore fact and stability oriented. However, stability is romantic. Any textual scholarship work must succumb to what could be called the "hermeneutic condition" – there exists nothing but interpretation.

When moving towards post-digital forms of interpretation, understanding the relation between hermeneutics, textual scholarship, and digital technology is an urgent and important academic need. An intimate understanding of digital code and how it affects textual scholarship is needed, but no structural scientific agenda to do so exists. In part this is the result of a lingering uneasiness between those scholars with a vested interest in digital (or "quantitative") approaches and those on the "qualitative" side of the spectrum. This methodological anxiety became more profound in the last two decades of the twentieth century and inhibited a sincere discourse concerning digital hermeneutics to develop within the digital humanities (DH). More recently some inroads seem to develop (cf. work by e.g. Rockwell, Stephen Ramsay, and Rafael Capurro). However, in debates there is still a tendency to reduce the potential of computation to a methodology of quantification and reduction (with, for instance, Johanna Drucker as a strong proponent of this argument). DH has a nature that is more hybrid than these debates suggest and there is not an a priori discontinuity with the hermeneutic traditions. Even if many computational methods in DH today lean towards quantified and reductive techniques, formalizations and analysis patterns

are not hermeneutics-free. Just as philological practice fundamentally cannot escape hermeneutics, neither modeling nor quantification can escape the hermeneutics involved in choosing the assumptions on which their formalizations are based, nor can they escape the hermeneutics involved with the interpretation of their results. Thus, as Katherine Hayles states, the tension between algorithmic analysis and hermeneutic close reading should not be overstated. However, a real challenge is, as inter alia David Berry and Federica Frabetti point out, that we need to understand the internal hermeneutics of code and computational methods to their core. We can only gain this understanding from a deep intimate engagement with "digitality" and software code itself.

Particular effects of the interaction between digital technology and textual scholarship do little to further this engagement. These effects may be uncovered using science and technology studies methods that offer a framework to systematically observe and reflect on scientific processes. In doing so, it appears that, for instance, a graphical user interface suggests a fictitious transparency of model and paradigm. The graphical interface is as much an opaque barrier to the internal paradigm of a system as it is a means of engaging with that very system. The interface is not the model, but a visualization that adds its own limitations, interpretations, and affordances. If interfaces are mostly formulated as expressions of an existing paradigm, they may well inhibit the appreciation and adoption of the affordances of new models and paradigms. This is for instance the case in textual scholarship where most digital editions are essentially digital remediations of books. They are based on a fully mimetic representational philosophy in which the digital edition is nothing more or less than a visualization that mimics as closely as possible the physical print document, which negates the affordances of other possible models. This effect I call paradigmatic regression, and it provides one explanation why extremely little experimentation with alternative modeling possibilities are found in textual scholarship. Such experimentation could even be based on very elementary and widely available Web technology. HTTP-links could be used to model intertextuality, for instance, which is a potential that has been pointed out well also within the textual scholarship community by, for example, George Landow and Theodor Nelson. Of course this paradigmatic effect is also linked to the conscience theoretical and practical choices that textual scholars make, both individually and as an academic community. The strong document-mimetic tenets of TEI-XML and the inviability of, for instance, the "social edition" suggest that these choices are in a certain sense conservative and in favor of a paradigm in which the ownership of the edition is still fully appropriated by the scholarly editor and in which the model is dominated by the format and limitations of the codex.

There is little evidence that the interaction between textual scholarship and digital technology results in significant methodological change. An expected effect of such change would be the development of a methodological pidgin and the exchange of technical and methodological vocabulary. The case studies presented in this thesis – mostly in the context of research at the Huygens Institute – do not provide examples of such exchange beyond some project management terminology that does not strictly relate to textual scholarship methods or computational analysis. What change there is seems to be introduced rather "covertly" at the level of programmed models and statistical analysis, the technical details of which are again covered over by graphical interfaces.

Paradigmatic regression effect is a divergent force that drives close and distant reading "schools" further apart rather than closer to each other. Scholarly editors and computational literary researchers supposedly cater to each other's needs. However, digital documentary editions are mostly computationally inaccessible monolithic XML-tag "cathedrals", while computational literary research projects produce mostly unsustainable "raw" text streams and annotations. A better common model is needed if the strengths of both need to apply to textual scholarship and digital scholarly editing. Knowledge graphs could provide such a model.

Arguably textual scholarship should take an interest in models of description and analysis that are more adequately geared towards digital objects. First of all because there is a general trend towards softwarization in society, resulting in ever more cultural objects and processes being expressed as digital objects and processes. Secondly because a similar softwarization process, more covertly than not, affects textual scholarship methods. If scholarly tasks and decisions are delegated to software engineers, and if many of these decisions and tasks are delegated by the programmers to the code that they develop, then it becomes utmost important that textual scholarship develops and adopts the means to make such delegated processes and decisions scientifically accountable in the case of bespoke code that is applied in scholarship. Contrary to what pernicious metaphors suggest, software is not a neutral and objective tool. Software code is a product of human creativity that harbors inbuilt assumptions about and models of reality. Thus, to warrant scientific process and quality control, a sufficient framework for code peer review and the critical study of code should be in place, so that engineers that create bespoke scholarly software can be held to scientific accountability and responsibility. The boundary between code that should be submitted to a form of code peer review and code that need not, is not exactly defined. Obviously the inner workings of a statistical computer language like R do not require scholarly peer review, but just the understanding that it has been rigorously examined and tested by computer scientists and statisticians. But code purposefully written – even if using, for instance, R – to execute a specific scholarly task in the context of a one-time-use research design, does require peer review. Notwithstanding the admirable achievements of software studies, critical code studies, and media-studies with regard to critical interrogating the role of software in society, such a framework for peer review of bespoke code applied in scholarship does not exist. Code peer review however, requires code literacy. And code literacy is a rare commodity in academic scholarship.

When code replaces certain tasks in scholarship this warrants questioning in some detail the role of those in scholarship that write the code. In a sense programmers are becoming scholarly authors and editors. Thus a discussion on authorship and editorship is relevant. Very concisely put the intellectual and philosophical history of authorship is a discourse on the question who "owns" the authority to determine the meaning of the text. Depending on time and context the answer to this question has varied between "deities" or "God" (in more historical times), "the author", and "the reader" (in more recent times). The claims to this authority over interpretation are related to claims on who gets to decide about truth. Obviously post-structuralist relativism and the "death of the author" recast any such claims as subjective.

Scholarly editing by and large responded in two ways to the post-structuralist "crisis" of factuality, determinability, and authority. One was by proposing archive-like and open forms of scholarly editions that fully recognize the intersubjectivity of editing and interpretation. This direction in scholarly editing recognizes the fluidity of text and the process-like nature of authoring, editing, and reading. It favors inclusivity of materials and readership. Ideally all possible resources related to a text should be included in an edition – which might rather be called "archive" in that case. The process of editing should be as open as possible and the edition itself should be open ended, so readers are able to add their own interpretations (e.g. by adding notes and annotations). Web 2.0 and digital technology in general are considered ideal means to realize this type of editions and editorial processes. The other response is a strong retreat on documentary editing. Rather than exploring new models, this direction in scholarly editing aims to impose the existing model of the book on digital technology. It reasserts and reaffirms the primacy of "philological fact" over interpretation. It is not uncommon for practitioners and theorists in textual scholarship to sway between these extremes, as for instance Jerome McGann did.

A closer inspection of the roles of author, editor, and engineer, reveals that writing code is a form of authorship. Both editing and programming in the context of textual scholarship turn out to be forms of revisionary authorship. However, in the majority of cases coding in a scholarly context is a form of unclaimed or, worse, misappropriated authorship. The authorship of code inserts at least one layer of interpretation into the process of scholarly editing in comparison to traditional modes of editing. This creates two problems. Firstly, the text and its interpretation potentially become even more fluid and unstable. This is contrary to the aim of philology and scholarly editing, which can generally be understood as an attempt to stabilize the text. Of course this notion of stabilizing the text is highly problematic in itself, but adding the authoring and performance of digital code into the process of editing is likely to further destabilize the text and its interpretation. One reason for this is that code is endowed with "deferred agency": its execution may result in autonomous scholarly decisions. Another reason is that code also has its own rhetoric and its own performativity, that are both still poorly understood. The second problem is that all possible effects of coding as a form of revisionary authorship within textual scholarship go unchecked by current scholarly processes. The reason for this is that programming is not recognized as authoring nor as scholarly editing, neither by scholars nor programmers. Therefore the potential influence of programming on scholarly editing, irrespective of its aims, goes largely unevaluated.

The introduction of the programmer – or computer science expert – into the scholarly process, raises an as of yet unanswered question of accountability. If digital objects and code are an integral part of the scholarly argument then their producers have an obligation to claim the contribution they make to the argument. Not just to make a righteous claim to academic credit, but foremost in order to be accountable for the scholarly argument they co-create. The intent of the revisionary authorship that code authoring produces, is not self-evident and not self-explanatory. This is true for any form of authorship. As early as the 1980s Donald Knuth acknowledged that code has rhetorical functions and that code authoring has poetics – if mostly only tacitly. He provided a model and implementation of a system to express the rhetoric and poetics of code more explicitly. Unfortunately his rationale has since been poorly understood and taught, and has been applied even less. Such rationales deserve revisiting and programmers should claim explicitly the revisionary authorship they appropriate through executable code.

The case studies in this dissertation show that an interdisciplinary community of developers and textual scholars is actively seeking new points of contact between scholarship and digital models of text. Different models foreground different aspects. TEI-XML seems in the majority of its applications to foreground the model of the codex and the structures of text in printed books. Statistical models in distant reading analyses foreground the idea of text as meaningfully quantifiable strings of tokens. Both models produce highly sophisticated and valuable interpretations. However, both models also put forward a particularly narrow understanding of texts as either predominantly document structures or "bags of words". They are demonstrable "lossy" with regard to the objects they describe, which is easily to gauge from the fact that they both forgo most material aspects of any text. This is not to say these models are fundamentally unable to capture and express these aspects, but rather to say that their particular makeup introduces a selective focus on the aspects they capture particularly well. If you use a "lens" that is exceptionally good at describing text structure, text structure is analytically foregrounded. If you use a lens that excels at describing bags of words, all texts start to look like bags of words. If you only have a hammer, many things look like nails.

The traditional analytical means of textual scholarship, to which for instance the classical apparatus belongs, are models too. Any model imparts some specific effect to interpretation. New models create new affordances that may lead to new perspectives and that may uncover new knowledge. As argued however, textual scholarship exhibits paradigmatic regression that inhibits productive experimentation with new models. The overwhelming freedom of modeling that the digital environment offers may be in part an explanation for this regression. That freedom stands in stark contrast to the specific constraints of epistemological devices such as an "apparatus criticus", that have been developed over multiple centuries and have been shaped to certain extents by the particular materiality of the codex. But in the interest of exploring new epistemologies for textual scholarship it would serve to also liberate texts from such known models and to examine the affordances of new and different models.

Case studies point in the direction of graphs as powerful means to express the multidimensionality of text. This suggests that the graph model is an interesting candidate to be one of these possible new models to be applied in textual scholarship. It can be succinctly shown that graph models offer a number of benefits over traditional models, such as feasibly solving TEI-XML overlap, representing multidimensionality, and computationally "mapping" textual variation.

Reflection on methodology is one of the most important obligations within any scientific domain. The social sciences, especially in the shape of science and technology studies, provide effective tools and techniques for such reflection, for instance through ethnography and autoethnography. Many more case studies of the interaction between computer science, software engineering, and the humanities will be needed before any firm conclusion can be drawn about what constitutes viable digital technological innovation in the latter, strongly hermeneutic domain. An autoethnography and case studies of attempts at such innovation in the textual scholarship domain at the Huygens Institute show that it is very difficult to innovate methodology while appreciating and safeguarding hermeneutic approach. It appeared that both scholars and technologists should strive to be much more aware of their own methodological limitations, and that both groups should accept the scientific strengths of the other group as complementing the weaknesses of their own group. In this respect, an autoenthnography of fifteen years experience of working on the intersection of these domains shows that both scholars and technologists need to be more respectfully aware of each others' scientific qualities, and that they need to be more aware of their individual and shared scientific responsibilities in methodological innovation, accountability, and reflection.

Current computational approaches to hermeneutic scholarship are too much rooted in the rhetoric of speed and scale of automation, driving attention away of a particular hard but interesting computational problem to explore: that of computational hermeneutics. Scholars should strive to be both less intimidated by, and more welcoming to computational explorations. While doing so they should be confident and fully aware of the immeasurable scientific wealth and value incurred by more than twenty centuries of hermeneutics.

Samenvatting

De hermeneutiek heeft een lange geschiedenis die in Europa onlosmakelijk is verbonden met de tekst- en editiewetenschap. Beiden vinden hun wortels uiteindelijk in de klassieke filosofie en de latere Bijbel-exegese. Zowel het postmodernisme met haar relativistische karakter als de sterke opkomst van digitale technologie – evenzo veranderlijk en onbestendig als invloedrijk – zijn twee 'disruptieve' krachten die inwerken op een editiewetenschap die zich bezighoudt met het culturele 'geheugen' van de mens en zijn cultuur en die daardoor gepreoccupeerd is met feiten en stabiliteit. Maar ook feiten en stabiliteit zelf zijn op hun best romantische idealen. Geen enkele vorm van editiewetenschap kan ontsnappen aan haar principiële hermeneutische karakter – er is niets dan interpretatie.

Nu een post-digitale werkelijkheid zich aandient, is een grondig begrip van de relatie tussen hermeneutiek, editiewetenschap en digitale technologie een belangrijke en urgente vereiste. Een diep begrip van de werking van softwarecode en hoe deze de editiewetenschap beïnvloedt is nodig, maar er is geen structurele wetenschappelijke agenda om dit probleem te bestuderen. Ten dele is dit het gevolg van de animositeit die in de laatste twee decennia van de twintigste eeuw tussen de 'kwantitatieven' en de kwalitatieve benadering van literatuur- en editiewetenschap ontstond en die de ontwikkeling van een serieus discourse omtrent digitale dan wel computationele hermeneutiek binnen de (digitale) geesteswetenschappen in de weg stond. Recentelijk zijn er wel aanwijzingen dat een serieuzer discourse zich kan ontwikkelen (Rockwell, Ramsay, Capurro), maar over het algemeen is er nog steeds een tendens om het potentieel van digitale en computationele technieken te reduceren tot een methodology van determinisme, reductivisme en kwantitatieve empiriek (Johanna Drucker is bij uitstek een voorbeeld van iemand die zulke perspectieven schetst). Het verwijt van reductivisme is snel gemaakt, maar het feitelijke karakter van digitale geesteswetenschappen is hybride en interdisciplinair, en er is niet a priori sprake van een breuk met de traditionele hermeneutiek. Zelfs al is het zo dat veel computationele methoden in de digitale geesteswetenschappen momenteel neigen naar reductieve en kwantificerende technieken, dan nog zijn hun formalizeringen en analyses allesbehalve vrij van hermeneutiek. Precies zoals de filologische praktijk fundamenteel niet kan ontsnappen aan interpretatie en daarmee aan hermeneutiek, kunnen ook digitale modelering en kwantificering niet ontsnappen aan de hermeneutiek die gemoeid is met de keuzes en aannames die ten grondslag liggen aan gekozen formaliseringen, en al helemaal niet aan de hermeneutiek die betrekking heeft op de interpretatie van analyseresultaten. We moeten – refererend aan de woorden van Katherine Hayles – de spanning tussen analyse met behulp van algoritmen en de hermeneutiek van "close reading" dan ook niet overdrijven. Maar het probleem is – zoals onder andere David Berry en Federica Frabetti – hebben aangegeven dat we de precieze innerlijke en impliciete hermeneutische werking van code en computationele technieken eigenlijk niet goed begrijpen. Dat doen we op dit moment met betrekking tot de tekst- en editiewetenschap zeker niet. Dat begrip kunnen we alleen kweken door een serieuze en langdurige omgang met en diepgravend onderzoek naar digitaliteit en software source-code in de context van tekst- en editiewetenschap.

Sommige effecten van de interactie tussen digitale technologie en de tekst- en editiewetenschap bevorderen het ontwikkelen van dat begrip niet. Die effecten kunnen we beter in beeld krijgen en begrijpen door middel van methoden uit de "Science and Technologie Studies" (STS). Deze bieden onder meer systematische technieken voor het observeren van en reflecteren op wetenschappelijke processen. Wanneer we dat doen, blijkt bijvoorbeeld dat grafische interfaces met name een fictieve transparantie van modellen en paradigma's bieden. Een grafische schil verhult een analytisch model even zozeer als dat hij dat model bruikbaar maakt. Een grafische schil is niet het model zelf, maar een visualisatie ervan met zijn eigen beperkingen, interpretaties en gebruiksmogelijkheden. Als grafische interfaces vooral worden vormgegeven als expressies van al bestaande en bekende paradigma's, modellen en metaforen – denk aan het "bureaublad" op een PC – kunnen ze een aanmerkelijke hindernis vormen voor het leren kennen en leren omgaan met nieuwe modellen, perspectieven en werkwijzen. Hiervan is bijvoorbeeld sprake in de editiewetenschap waar verreweg de meeste digital edities in essentie digitale metaforen van het boek zijn. Deze volgen een logica die volledig boek-mimetische is en waarin de digitale editie niets meer of minder is dan een zo nauwkeurig mogelijke imitatie van het fysieke document. Deze benadering negeert vrijwel alle nieuwe intrinsieke omgangsvormen met het tekstmateriaal die andere digitale modellen mogelijk bieden. Dit effect heb ik "paradigmatische regressie" genoemd, en het is één van de redenen waarom we – in de context van tekst- en editiewetenschap – weinig experimenten met alternatieve modellen voor digitale tekst aantreffen. Zulke experimenten zouden zelfs al gebaseerd kunnen zijn op de toepassing van een zeer elementaire en wijdverspreide Internet-technologie zoals de http-link. Bijvoorbeeld voor het beschrijven en modelleren van intertekstualiteit. Deze mogelijkheid mag ook binnen de gemeenschap van tekst- en editiewetenschappen – bij monde van bijvoorbeeld Theodor Nelson en George Landow – genoegzaam bekend verondersteld worden, maar tot geïnspireerde experimentele digitale edities heeft dit niet geleid. Natuurlijk is het regressie-effect niet alleen een gevolg van technologie. Het volgt ook uit bewuste theoretische en praktijkmatige keuzes door individuele editeurs en van keuzes binnen hun academische gemeenschap. De sterke neiging van TEI-XML tot het imiteren van documenten en de levenloos geboren "social edition" suggereren dat deze keuzes in zekere zin conservatief zijn en in lijn met de idee dat het mandaat van editeren en editie nog altijd volledig geclaimd wordt door de wetenschappelijk editeur en een tekstmodel nog steeds geheel is onderworpen aan de vorm en beperkingen van het fysieke boek.

Er is weinig bewijs te vinden dat de interactie tussen editiewetenschappers en digitale technologie resulteert in belangwekkende technisch-methodologische innovatie. Een verwacht effect zou de ontwikkeling van een methodisch-technische mengtaal zijn, een methodologisch pidgindialect dat duidt op de uitwisseling van terminologie en werkwijzen tussen de domeinen. De case studies in deze dissertatie, die met name dergelijk ontwikkelingen beschouwen binnen de context van het Huygens Instituut, getuigen niet van dergelijke uitwisseling behoudens enige terminologie uit het IT-projectmanagement die niet strikt betrekking heeft op tekst- en editiewetenschappelijke methodologie of op computationele methoden. De methodologische veranderingen waarnaar gewezen kan worden lijken eerder nogal onzichtbaar te worden doorgevoerd op het niveau van programmatuur, softwarematige modellen en statistische methoden, waarvan de technische details opnieuw worden toegedekt door grafische interfaces en visualiseringen.

Het paradigmatisch regressie-effect is een kracht die de scholen van "close" en "distant reading" eerder uit elkaar dan naar elkaar toe drijft. Terwijl in theorie XML-gebaseerde edities ook in dienst staan van computationeel literair onderzoek, zijn het in de praktijk onontwarbare kluwen van XML-tags. Van de andere kant hebben computationeel literair analisten een voorkeur voor het 'on the fly' annoteren en analyseren van niet-onderhoudbare digitale wegwerpteksten. Een gedeeld tekstmodel c.q. formaat is noodzakelijk als de expertise en kunde van beide domeinen elkaar blijvend willen versterken. Kennisgrafen ("knowledge graphs") lijken hier uitkomst te bieden.

Het lijkt vanzelfsprekend dat de editiewetenschappen vanuit principe geïnteresseerd zouden moeten zijn in betere modellen voor de beschrijving an analyse van digitale objecten (zoals digitale teksten en edities). Ten eerste omdat er een algemene trend van digitalisering van cultuur en maatschappij gaande is waardoor culturele objecten en processen steeds vaker digitale objecten en processen zijn. Ten tweede omdat deze 'softwarisatie' ook zijn weerslag heeft op de methodes van de editietechniek, zij het meer onzichtbaar dan zichtbaar. Wanneer geesteswetenschappelijke activiteit en tekstkritische beslissingen worden gedelegeerd aan softwareontwikkelaars en deze door softwareontwikkelaars opnieuw gedelegeerd worden naar de code die zij schrijven, dan is het belangrijk dat de editiewetenschap beschikt over methoden en technieken om zulke gedelegeerde processen en taken te toetsen op wetenschappelijke correctheid. Dit is van des te groter belang naarmate ontwikkelde software specifiek geschreven is voor eenmalige analytische toepassing (zogeheten "bespoke" of "tailor-made" code). In tegenstelling tot wat gebruikelijke maar misplaatste metaforen willen doen geloven, is softwarecode geen neutrale amorele techniek, maar een product van menselijke creativiteit met ingebedde aannames en perspectieven op de realiteit. Dus om de wetenschappelijkheid en kwaliteit van het editieproces te waarborgen dient er een voldoende systematiek te zijn voor softwarekritiek en voor de "peer review" van softwarecode. Op deze wijze kan de wetenschappelijke verantwoordelijkheid en ethiek van softwareontwikkelaars die dergelijk code produceren bevorderd en gewaarborgd worden. Het is momenteel niet strikt eenduidig vast te stellen welke categorieën van code onder een dergelijke regime zouden moeten vallen en welke niet. Het is evident dat een computertaal zoals R niet nog eens door tekstwetenschappers gecontroleerd hoeft te worden – zo zij dit al zouden kunnen. Zo'n techniek is ruim voldoende geëvalueerd door computerwetenschappers en statistici. Maar software die specifiek geschreven is – bijvoorbeeld in R – om een specifieke editietaak uit te voeren in de context van een specifiek en éénmalig editieproject, zou wel degelijk aan code peer review onderhevig moeten zijn als onderdeel van de wetenschappelijke evaluatie. Niettegenstaande het fascinerende werk dat al gebeurt in softwarestudies, ccs ("critical code studies") en mediastudies met betrekking tot de kritische bestudering van de rol van software in cultuur en maatschappij, is er geen in het wetenschappelijk proces ingebedde systematiek voor code peer review in de tekst- en editiewetenschappen, noch in de geesteswetenschappen in brede zin. Voor code peer review is het kunnen lezen en begrijpen van code echter noodzakelijk en deze vaardigheden zijn zeldzaam in de geesteswetenschappen in het algemeen en in de editiewetenschap in het bijzonder.

Wanneer bepaalde editiewetenschappelijke taken gedelegeerd worden aan softwarecode is het opportuun de rol van diegenen die die softwarecode schrijven nader te beschouwen. In een zekere zin worden programmeurs ook wetenschappelijk auteurs en editeurs, wat een beschouwing omtrent auteurschap en editeurschap relevant maakt. Extreem verkort kan de intellectuele en filosofische geschiedenis van het auteurschap gezien worden als een discourse omtrent wie het mandaat heeft op het definiëren van de betekenis van tekst. Afhankelijk van tijd en context heeft het antwoord op die vraag gevarieerd van "goden" en "God" in vroeger tijden tot "de auteur" en uiteindelijk "de lezer" in meer recente tijden. Het claimen van een dergelijk mandaat is gerelateerd aan het claimen van zekere (soms absolute) waarheden. Postmodern relativisme en de doodverklaring van de auteur door onder andere Derrida hebben duidelijk gemaakt dat dit soort claims altijd (inter)subjectief zijn.

De editiewetenschap heeft grosso modo op twee verschillende manieren gereageerd op het postmoderne relativisme en de daaruit volgende 'crisis' met betrekking tot feit en waarheid. Eén manier is om de editie voor te stellen als een vorm van archief waarbij alle relevante bronnen en interpretaties betrokken kunnen en idealiter moeten worden. Deze richting accepteert volledig de intersubjectiviteit van interpretatie en het procesmatige karakter van schrijven, lezen en editeren, alsmede het 'fluïde' karakter van tekst. Deze benadering is inclusief met betrekking tot oorspronkelijk materiaal maar ook met betrekking tot de lezer c.q. gebruiker, die uitgenodigd wordt om interpretaties en annotaties toe te voegen, waardoor een 'open science' en 'open einde' editie ontstaat. Deze richting in het editeren wijst dan ook graag naar digitale (Web)technologie om deze open vormen van interactiviteit te realiseren. De andere reactie is meer reactionair en streeft er naar het model van het boek opnieuw als de dominante vorm te bevestigen. Deze stroming streeft ernaar de "filologische feiten" voorrang te verlenen boven een hegemonie van interpretatie en fluïditeit. Wat betreft hun eigen voorkeur is het niet ongebruikelijk voor editeurs om zowel in theorie als in de praktijk van het editeurschap heen en weer geslingerd te worden tussen deze extreme posities, zoals bijvoorbeeld in het geval van Jerome McGann.

Uit nadere beschouwing van de rollen van auteur, editeur en softwareontwikkelaar blijkt dat het schrijven van code ook een vorm van auteurschap is. In de context van tekst- en editiewetenschappen zijn zowel editeur als programmeur in wezen revisionaire auteurs van oorspronkelijke werken. Maar in verreweg de meeste gevallen blijkt het schrijven van code in het geval van digitale edities niet geclaimd te worden als een vorm van auteurschap – vaak wordt alleen de claim van de editeur op het werk erkend. In vergelijking met traditionele vormen van editiewetenschap voegt het schrijven van code tenminste één laag van interpretatie toe aan het proces van editeren. Dit introduceert in ieder geval twee nieuwe problemen. Ten eerste worden tekst en interpretatie nog meer fluïde en daarmee meer instabiel, wat ingaat tegen wat nog vaak als doel van filologie en tekstwetenschap wordt gezien: het op een zekere manier stabiliseren van de tekst. Natuurlijk is het idee van de stabiele tekst an sich al erg problematisch, maar de toevoeging van het schrijven en toepassen van softwarecode creëert hoogstens meer mogelijkheden om de tekst en zijn interpretatie op verschillende manieren te veranderen en te vermeerderen en daarmee te destabiliseren. Temeer omdat code door ingebouwde heuristiek in zekere zin gedelegeerde (wetenschappelijke) activiteit kan verrichten, en in bepaalde gevallen autonome beslissingen neemt. Daarbij heeft code een retoriek en performativiteit die tot nu toe weinig systematisch onderzocht is en dus weinig begrepen. Daarbij is het Belangrijk om te constateren dat geen van deze effecten effectief gecontroleerd wordt als onderdeel van het (editie)wetenschappelijk proces omdat programmeren door noch ontwikkelaar noch geesteswetenschapper erkend wordt als schrijven of als editeren. Hierdoor blijft de invloed van programmeren op de editiewetenschappelijke methodologie grotendeels onbekend en ongeëvalueerd.

De introductie van de programmeur – of computer science expert, ontwikkelaar, IT-specialist, of hoe degene ook wordt genoemd die de code schrijft waarvan gezegd kan worden dat deze een geesteswetenschappelijke activiteit uitvoert – in het geesteswetenschappelijk proces leidt tot een tot nog toe onopgelost probleem van wetenschappelijke verantwoordelijkheid. Wanneer digitale objecten en softwarecode integraal onderdeel worden van een geesteswetenschappelijke argumentatie, dan dienen de producenten van die originele bijdrage een claim te leggen op de creatie ervan. Niet alleen om gerechtvaardigd claim te leggen op de academische waarde en waardering voor dat werk, maar vooral om de wetenschappelijke verantwoordelijkheid te nemen voor de wetenschappelijke argumentatie die zij zo co-creëren. De intentie en betekenis van het revisionaire auteurschap dat code produceert spreekt niet voor zich. Dit houdt voor elke vorm van auteurschap. Al in de jaren 1980 beargumenteerde Donal Knuth dat code retorische functies heeft en dat programmeurs een, zij het vaak impliciete, poëtica hanteren. Hij produceerde zelfs een model en een implementatie van een computertaal om deze aspecten expliciet te kunnen maken. Helaas is zijn betoog en motivering later meesttijds verkeerd begrepen, weinig tot niet onderwezen en al helemaal zelden toegepast. Maar deze meer fundamentele theoretisch retorische beginselen van code verdienen hernieuwde aandacht als beginselen van het revisionaire auteurschap van code dat programmeurs vaker en actiever zouden moeten claimen.

De case studies in deze dissertatie laten zien dat een interdisciplinaire gemeenschap van ontwikkelaars en tekstwetenschappers actief op zoek is naar de mogelijk interacties tussen geesteswetenschappelijke analyse en digitale modellen voor tekst. Verschillende modellen benadrukken verschillende aspecten van datgene wat ze modelleren. TEI-XML lijkt in de meeste gevallen het model van de codex (boek) en de tekststructuur van fysiek gedrukte werken op de voorgrond te plaatsen. De statistische modellen uit de "distant reading" methodologie benaderen een tekst als een reeks van tekens en tokens en benadrukken de meetbaarheid daarvan. Beide modellen stellen onderzoekers in staat tot hoogwaardige en verfijnde analyse en interpretatie. Maar beide gaan ook uit van een erg nauwe en armoedige opvatting van wat tekst is: structuren of strings. Ze verliezen beide aantoonbaar veel informatie ten aanzien van elke fysiek object dat ze beschrijven, wat eenvoudig te demonstreren valt door te wijzen op het feit dat ze bijvoorbeeld veelal alle materiële eigenschappen van een fysieke bron negeren. Dat wil niet zeggen dat die modellen die aspecten fundamenteel niet kunnen representeren, maar dat hun specifieke compositie ze bijzonder geschikt maak om bepaalde aspecten te beschrijven waardoor zij die aspecten uitlichten en benadrukken. Wanneer je een 'lens' toepast die bijzonder geschikt is om tekststructuur bloot te leggen, dan zal tekststructuur analytisch gezien op de voorgrond geplaatst worden. Als je een 'lens' gebruikt die een tekst als een losse verzameling woorden beschrijft, dan zullen alle teksten er als losse verzamelingen van woorden uit gaan zien. Als je alleen een hamer hebt, zien verdacht veel dingen eruit als spijkers.

De conventionele analytische gereedschappen van de tekst- en editiewetenschap, waartoe bijvoorbeeld het "apparatus criticus" behoort, zijn ook modellen. Deze modellen geven hun eigen specifieke focus aan interpretatie en analyse. Nieuwe en andere modellen introduceren nieuwe mogelijkheden, perspectieven en uiteindelijk kennis. Zoals uitgelegd vertoont de editiewetenschap een zekere mate van paradigmatische regressie die het haar lastig maakt veel met nieuwe modellen te experimenteren. Een deel van de verklaring voor deze relatieve neiging tot regressie moet gezocht worden in de overweldigende vrijheid van modelleren die digitale technologie biedt. Die bijna absolute vrijheid staat in groot contrast tot de specifieke en bewust opgelegde en gehanteerde beperkingen van het "apparatus criticus" dat over verscheidene eeuwen heen ontwikkeld is, en mede is ingegeven en vormgeven door de grenzen en beperkingen van het fysieke boek. Maar met het oog op het onderzoeken van een nieuwe digitale of computationele epistemologie zou het nuttig zijn om tekst ook soms te bevrijden van deze welbekende modellen en de epistemologische mogelijkheden van meer en andere modellen voor tekst te onderzoeken.

Een aantal case studies wijst in de richting van grafen als een krachtige expressie van de multidimensionale aspecten van tekst. Zulke grafen zouden dus een goede kandidaat kunnen zijn voor een nieuw tekstmodel in de editiewetenschap en tekstkritiek. De voorbeelden in de case studies tonen aan dat zulke graafgebaseerde modellen specifieke voordelen bieden boven conventionele modellen – bijvoorbeeld een praktische oplossing voor het probleem van overlap in TEI-XML, de representatie van multidimensionaliteit, en het inzichtelijk in kaart brengen van tekstvariatie.

Een belangrijke wetenschappelijke verantwoordelijkheid voor elk wetenschappelijk veld is de routinematige reflectie op de eigen methodologie. De sociale wetenschappen, en meer speciaal de Science and Technology Studies als onderdeel daarvan, beschikken over uitstekende methoden en technieken voor dergelijke reflectie. Bijvoorbeeld in de vorm van etnografie en auto-etnografie. Voordat er generaliserende conclusies getrokken kunnen worden omtrent waardevolle wijzen van digitaal-technologische innovatie in het hermeneutisch domein zullen er nog veel case study gebaseerde onderzoeken nodig zijn naar de interactie tussen computerwetenschap, softwareontwikkeling en de geesteswetenschappen. Etnografisch en auto-etnografisch onderzoek naar pogingen tot zulke innovatie in de editiewetenschap in de context van het Huygens Instituut toont aan dat het erg moeilijk is om methodologie in dit domein te innoveren op basis van digitale methoden met zorgvuldig behoud van de hermeneutische benadering. Het blijkt dat zowel geesteswetenschappers als digitaal specialisten en ontwikkelaars ernaar moeten streven zich meer bewust te zijn van de eigen methodologische beperkingen en meer zouden moeten proberen om elkaars wetenschappelijke sterke punten te waarderen als aanvulling op de eigen methoden en technieken. Een auto-etnografie van vijftien jaar ervaring op het kruispunt van deze wetenschappelijke domeinen laat zien dat zowel geesteswetenschappers en digitaal-technologische specialisten zich meer respectvol bewust en geïnteresseerd kunnen tonen voor elkaars wetenschappelijke kwaliteiten, en dat beide zich bewuster kunnen zijn van hun individuele en gemeenschappelijke wetenschappelijke verantwoordelijkheid ten aanzien van methodologische reflectie en innovatie.

De huidige computationele benaderingen die worden toegepast in het hermeneutische domein zijn behept met de retoriek van snelheid en schaalvergroting uit de industriële automatisering. Deze retoriek en de bijbehorende benaderingen creëren een blinde vlek voor een belangrijke en echte computationele uitdaging: dat van het probleem van de computationele hermeneutiek. Computationeel onderzoekers zouden aangemoedigd moeten worden deze uitdaging aan te gaan. Van de andere kant doen geesteswetenschappers er goed aan om te pogen minder geïntimideerd te raken door hun computationele collega's en deze computationele ontdekkingstocht aan te gaan vanuit het vertrouwen en het volle besef dat hun werk en methoden wortelt in de onmetelijk waardevolle wetenschappelijk kracht van meer dan twintig eeuwen hermeneutiek.

Bibliography

AAP. 2010. “AAP Reports October Book Sales.” AAP The Association of American Publishers. December 8, 2010. http://www.winterdigital.com/work/AAP_final/main/PressCenter/Archicves/2010_Dec/AAPReportsOctoberBookSales.htm.

Ackoff, R.L. 1989. “From Data to Wisdom: Presidential Address to ISGSR, June 1988.” *Journal of Applied Systems Analysis* 16: 3–9.

Akker, van den, Chiel, Susan Legêne, van, Marieke Erp, van, Lora Aroyo, Roxane Segers, Lourens Meij, van der, Jacco Ossenbruggen, et al. 2011. “Digital Hermeneutics: Agora and the Online Understanding of Cultural Heritage.” In *Proceedings of the ACM WebSci’11*, 1–7. Koblenz. http://www.websci11.org/fileadmin/websci/Papers/116_paper.pdf.

Alvarado, Rafael C. 2012. “The Digital Humanities Situation.” In *Debates in the Digital Humanities*, edited by Matthew K. Gold, 50–55. Minneapolis: University of Minnesota Press. http://dhdebates.gc.cuny.edu/debates/text/50.

Anderson, Leon. 2006. “Analytic Autoethnography.” *Journal of Contemporary Ethnography* 35 (4): 373–95. https://doi.org/10.1177/0891241605280449.

Andrews, Tara L., and Caroline Macé. 2013. “Beyond the Tree of Texts: Building an Empirical Model of Scribal Variation through Graph Analysis of Texts and Stemmata.” *Literary and Linguistic Computing* 28 (4): 504–521. https://doi.org/10.1093/llc/fqt032.

Andrews, Tara L., and Joris J. Van Zundert. 2018. “What Are You Trying to Say? The Interface as an Integral Element of Argument.” In *Digital Scholarly Editions as Interfaces*, edited by Roman Bleier, Martina Bürgermeister, Helmut W. Klug, Frederike Neuber, and Gerlinde Schneider, 3–33. Schriften Des Instituts Für Dokumentologie Und Editorik 12. Norderstedt: Books on Demand.

Andrews, Tara L., Joris Job Van Zundert, Thomas Efer, Stefan Jänicke, Andreas Kuczera, and Stefan Dumont. 2018. “Graph Models for Textual Data: Between Text and Information.” In *EADH 2018: “Data in Digital Humanities,”* 31. Galway: National University of Ireland. https://eadh2018.exordo.com/files/papers/19/initial_draft/TAG_panel_-_Google_Docs.pdf.

Antonijević, Smiljana. 2015. *Amongst Digital Humanists: An Ethnographic Study of Digital Knowledge Production*. London, Houndmills, New York: Palgrave Macmillan.

Baecker, Ronald M., Jonathan Grudin, William A.S. Buxton, and Saul Greenberg, eds. 1995. *Readings in Human-Computer Interaction: Toward the Year 2000*. Second Edition. San Francisco: Morgan Kaufmann Publishers, Inc.

Barabucci, Gioele, Angelo Di Iorio, and Fabio Vitali. 2014. "Stemma codicum: analisi e generazione semi-automatica." In *Digital Humanities: progetti italiani ed esperienze di convergenza multidisciplinare: Atti del convegno annuale dell'Associazione per l'Informatica Umanistica e la Cultura Digitale (AIUCD). Firenze, 13-14 dicembre 2012*, edited by Fabio Ciotti, 24:129–145. Studi umanistici, serie Quaderni DigiLab. Firenze: Sapienza Università Editrice.

Barolini, Teodolinda. 2007. "Introduction." In *Petrarch and TheTextual Origins of Interpretation*, edited by H. Wayne Storey and Teodolinda Barolini, 31:1–12. Columbia Studies in the Classical Tradition. Leiden, Boston: Brill.

Barthes, Roland. 1967. "The Death of the Author." Edited by Brian O'Doherty. *Aspen* 5+6. http://www.ubu.com/aspen/aspen5and6/threeEssays.html#barthes.

———. 1975. *S/Z: An Essay*. Translated by Richard Miller. New York: Hill and Wang.

———. 1989. *Sade / Fourier / Loyola*. Translated by Richard Miller. Berkeley, Los Angeles: University of California Press.

Bauer, Jean. 2011. "Who You Calling Untheoretical?" *Journal of Digital Humanities* 1 (1). http://journalofdigitalhumanities.org/1-1/who-you-calling-untheoretical-by-jean-bauer/.

Beaulieu, Anne, Karina Van Dalen-Oskam, and Joris Van Zundert. 2012. "Between Tradition and Web 2.0: ELaborate as a Social Experiment in Humanities Scholarship." In *Social Software and the Evolution of User Expertise: Future Trends in Knowledge Creation and Dissemination*, edited by Tatjana Takševa, 112–129. Hershey: IGI Global. https://10.4018/978-1-4666-2178-7.ch007.

Berners-Lee, Tim. 1989. "Information Management: A Proposal." CERN. http://info.cern.ch/Proposal.html.

Berry, David. 2013. "Post-Digital Humanities." *Stunlaw, A Critical Review of Politics, Arts and Technology*. October 18, 2013. http://stunlaw.blogspot.nl/2013/10/post-digital-humanities.html.

Berry, David M. 2012. "Introduction: Understanding the Digital Humanities." In *Understanding Digital Humanities*, edited by David M. Berry, 1–20. Basingstoke, New York: Palgrave Macmillan.

———. 2014. *Critical Theory and the Digital.* Critical Theory and Contemporary Society. New York, London, New Delhi etc.: Bloomsbury Academic.

Bijker, Wiebe E., Thomas P. Hughes, and Trevor J. Pinch. 1987. *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology.* Cambridge: MIT Press.

Bishop, Dorothy. 2012. "How to Bury Your Academic Writing." *BishopBlog.* August 26, 2012. http://deevybee.blogspot.co.uk/2012/08/how-to-bury-your-academic-writing.html.

Blake, William. 1996. *Poems of William Blake.* Project Gutenberg. http://www.gutenberg.org/ebooks/574.

Bod, Rens. 2013a. *A New History of the Humanities: The Search for Principles and Patterns from Antiquity to the Present.* Oxford: Oxford University Press.

———. 2013b. "Het Einde van de Geesteswetenschappen 1.0." Vossiuspers UvA. http://staff.science.uva.nl/~rens/OratieRens.pdf.

———. 2013c. "Who's Afraid of Patterns? The Particular versus the Universal and the Meaning of Humanities 3.0." *BMGN – Low Countries Historical Review* 128 (4): 171–180. https://doi.org/10.18352/bmgn-lchr.9351.

Bogost, Ian. 2015. "Programmers: Stop Calling Yourselves Engineers." *The Atlantic*, November 5, 2015. http://www.theatlantic.com/technology/archive/2015/11/programmers-should-not-call-themselves-engineers/414271/.

Bolter, Jay David, and Richard Grusin. 2000. *Remediation: Understanding New Media.* Cambridge: The MIT Press.

Boot, Peter, and Joris Zundert van. 2011. "The Digital Edition 2.0 and The Digital Library: Services, Not Resources." *Bibliothek Und Wissenschaft* 44: 141–152. http://peterboot.nl/pub/vanzundert-boot-services-not-resources-2011.pdf.

Bordalejo, Barbara. 2018. "Digital versus Analogue Textual Scholarship or The Revolution Is Just in the Title." *Digital Philology: A Journal of Medieval Cultures* 7 (1): 7–28. https://doi.org/10.1353/dph.2018.0001.

Borgman, Christine L. 2007. *Scholarship in the Digital Age: Information, Infrastructure, and the Internet.* Cambridge, London: The MIT Press.

———. 2009. "The Digital Future Is Now: A Call to Action for the Humanities." *Digital Humanities Quarterly* 3 (4). https://www.digitalhumanities.org/dhq/vol/3/4/000077/000077.html.

———. 2015. *Big Data, Little Data, No Data: Scholarship in the Networked World.* Cambridge, London: The MIT Press.

Bowker, Geoffrey C. 2006. *Memory Practices in the Sciences.* Cambridge, London: MIT Press.

Bray, Tim, and C.M. Sperberg-McQueen. 1996. "Extensible Markup Language

(XML)." W3C Working Draft. W3C. November 14, 1996. http://www.w3.org/TR/WD-xml-961114.html.

Bridenbaugh, Carl. 1963. "The Great Mutation." *American Historical Review* 68 (2): 315–331. http://www.historians.org/info/AHA_History/cbridenbaugh.htm.

Brumfield, Ben. 2012. "Crowdsourcing at IMLS WebWise 2012." *Collaborative Manuscript Transcription*. March 17, 2012. http://manuscripttranscription.blogspot.nl/2012/03/crowdsourcing-at-imls-webwise-2012.html.

———. 2013. "The Collaborative Future of Amateur Editions." *Collaborative Manuscript Transcription*. July 13, 2013. http://manuscripttranscription.blogspot.com/2013/07/the-collaborative-future-of-amateur.html.

Brumfield, Ben, David Klevan, and Ben Vershbow. 2012. "Sharing Public History Work Using Crowdsourcing of Both Data and Sources." Panel presented at the Webwise 2012, Baltimore, February 29. http://www.imls.gov/about/webwise.aspx.

Brüning, Gerrit, Katrin Henzel, and Dietmar Pravida. 2013. "Multiple Encoding in Genetic Editions: The Case of 'Faust'." *Journal of the Text Encoding Initiative*, no. 4 (March). https://doi.org/10.4000/jtei.697.

Bryant, J. 2002. *The Fluid Text: A Theory of Revision and Editing for Book and Screen*. University of Michigan Press. http://books.google.nl/books?id=1w4wpOdPbu4C.

Buckland, Michael. 1997. "What Is a 'Document'?" *Journal of the American Society of Information Science* 48 (9): 804–809. http://people.ischool.berkeley.edu/~buckland/whatdoc.html.

Bulhak, Andrew C. 1996. "On the Simulation of Postmodernism and Mental Debility Using Recursive Transition Networks." Department of Computer Science, Monash University. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.4137&rep=rep1&type=pdf.

Bullynck, Maarten. 2016. "Histories of Algorithms: Past, Present Ad Future." *Historia Mathematica* 43 (3): 332–341. https://doi.org/10.1016/j.hm.2015.07.002.

Burgess, Helen J., and Jeanne Hamming. 2011. "New Media in Academy: Labor and the Production of Knowledge in Scholarly Multimedia." *DHQ: Digital Humanities Quarterly* 5 (3). http://digitalhumanities.org/dhq/vol/5/3/000102/000102.html.

Burke, John M. 1989. "The Death and Return of the Author: Criticism and Subjectivity in Barthes, Foucault and Derrida." PhD, Edinburgh: University of Edinburgh. https://www.era.lib.ed.ac.uk/handle/1842/7375.

Burke, Seán, ed. 1995. *Authorship: From Plato to the Postmodern—a Reader*. Edinburgh: Edinburgh University Press.

Burnard, Lou. 1998. "On the Hermeneutic Implications of Text Encoding." In *New Media in the Humanities: Research and ApplicationsProceedings of the First Seminar on Computers, Literature and Philology*, edited by Domenico Fiormonte and John Usher, 39–45. Edinburgh. http://users.ox.ac.uk/~lou/wip/herman.htm.

Burrows, John. 2002. " 'Delta': A Measure of Stylistic Difference and a Guide to Likely Authorship." *Literary & Linguistic Computing* 17 (3): 267–287. https://doi.org/10.1093/llc/17.3.267.

Buschenhenke, Floor. 2016. "Het literaire werk anno 2016: Digitale schrijfprocessen vastleggen en analyseren." *Vooys* 34 (4): 8–20. https://pure.knaw.nl/portal/files/7195159/34.4Buschenhenke_artikelVooys.pdf.

Bush, Vannevar. 1945. "As We May Think." *The Atlantic*, July 1, 1945. http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/.

Buzzetti, D. 2002. "Digital Representation and the Text Model." *New Literary History* 33 (1): 61–88.

———. 2012. "AIUCD Conference." Email to author. November 10, 2012.

Buzzetti, Dino. 2009. "Digital Editions and Text Processing." In *Text Editing, Print and the Digital World*, edited by Marilyn Deegan and Kathryn Sutherland, 45–61. Farnham, Burlington: Ashgate. http://www.academia.edu/391823/Digital_Editions_and_Text_Processing.

Buzzetti, Dino, and Jerome McGann. 2006. "Critical Editing in a Digital Horizon." In *Electronic Textual Editing*, edited by Lou Burnard, Katherine O'Brien O'Keeffe, and John Unsworth, 53–73. New York: Modern Language Association of America. http://www.tei-c.org/Activities/ETE/Preview/index.xml.

Buzzetti, Dino, and Malte Rehbein. 1998. "Textual Fluidity and Digital Editions." In *Text Variety in the Witnesses of Medieval Texts*, edited by Milena Dobreva, 14–39. Sofia: Institute of Mathematics and Informatics. http://www.denkstaette.de/files/Buzzetti-Rehbein.pdf.

Cain Miller, Claire, and Julie Bosman. 2011. "E-Books Outsell Print Books at Amazon." *New York Times*, May 19, 2011, sec. Technology. http://www.nytimes.com/2011/05/20/technology/20amazon.html?_r=0.

Camps, Jean-Baptiste, and Florian Cafiero. 2014. "Genealogical Variant Locations and Simplified Stemma: A Test Case." In *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, edited by Tara Andrews and Caroline Macé, 69–94. Lectio 1. Turnhout: Brepols. https://doi.org/10.1484/M.LECTIO-EB.5.102565.

Capurro, Rafael. 2010. "Digital Hermeneutics: An Outline." *AI & Society* 35 (1): 35–42. http://www.capurro.de/digitalhermeneutics.html.

Cardinali, Lucilla, Francesca Frassinetti, Claudio Brozzoli, Christian Urquizar, Alice C. Roy, and Alessandro Farnè. 2009. "Tool-Use Induces Morphological Updating of the Body Schema." *Current Biology* 19 (12): R478–479. http://www.sciencedirect.com/science/article/pii/S0960982209011099.

Carusi, Annamaria. 2011. "Transformational Humanities: Beyond User Requirements in the Digital Humanities." Lecture presented at the Launch of Alfalab, Amsterdam, September 29. http://alfalablog.huygens.knaw.nl/?p=608.

Causer, T., J. Tonra, and V. Wallace. 2012. "Transcription Maximized; Expense Minimized? Crowdsourcing and Editing The Collected Works of Jeremy Bentham." *Literary and Linguistic Computing* 27 (2): 119–137. https://doi.org/10.1093/llc/fqs004.

Causer, Tim, and Melissa Terras. 2014. "Crowdsourcing Bentham: Beyond the Traditional Boundaries of Academic History." *International Journal of Humanities and Arts Computing* 8 (1): 46–64. http://discovery.ucl.ac.uk/1354678/.

Cerquiglini, Bernard. 1999. *In Praise of the Variant: A Critical History of Philology*. Baltimore: The Johns Hopkins University Press.

Ceruzzi, Paul E. 2012. *Computing: A Concise History*. Cambridge, London: MIT Press.

Chambers, Ellie. 2000. "Editorial: Computers in Humanities Teaching and Research." *Computers and the Humanities* 34 (3): 245–254. http://www.jstor.org/stable/30204817.

Chandra, Vikram. 2014. "The Beauty of Code." The Paris Review Magazine Website. *The Daily*. September 5, 2014. http://www.theparisreview.org/blog/2014/09/05/the-beauty-of-code/.

Ciccoricco, David. 2010. "Individual Work: Days of JavaMoon." Electronic Literature Directory. August 12, 2010. https://directory.eliterature.org/individual-work/617.

Ciula, Arianna, and Cristina Marras. 2016. "Circling around Texts and Language: Towards 'Pragmatic Modelling' in Digital Humanities." *DHQ: Digital Humanities Quarterly* 10 (3). http://www.digitalhumanities.org/dhq/vol/10/3/000258/000258.html.

Coffee, Neil, Jean-Pierre Koenig, Shakthi Poornima, Roelant Ossewaarde, Richard Forstall, and Sarah Jacobson. 2012. "Intertextuality in the Digital Age." *Transactions of the American Philological Association* 42 (2): 383–422. https://doi.org/10.1353/apa.2012.0010.

Coffey, Amanda. 1991. *The Ethnographic Self: Fieldwork and the Representation of*

Identity. London, Thousand Oaks, New Dehli: SAGE Publications.

Cohen, Shay. 2016. *Bayesian Analysis in Natural Language Processing*. Synthesis Lectures on Human Language Technologies 35. Morgan & Claypool Publishers.

Compagnon, Antoine. 2004. *Literature, Theory, and Common Sense*. Translated by Carol Cosman. Princeton and Oxford: Princeton University Press.

Courant, Paul N., Sarah E. Fraser, Michael Goodchild, Margaret Hedstrom, Charles Henry, Peter B. Kaufman, Jerome McGann, John Unsworth, and Bruce Zuckerman. 2006. *Our Cultural Commonwealth: The Report of the American Council of Learned Societies' Commission on Cyberinfrastructure for Humanities and Social Sciences*. Edited by Marlo Welshons. New York: American Council of Learned Societies.

Coyne, Richard. 1995. *Designing Information Technology in the Postmodern Age: From Method to Metaphor*. Leonardo Book Series. Cambridge: The MIT Press.

Crombie, Alistair Cameron. 1995. *Styles of Scientific Thinking in the European Tradition : The History of Argument and Explanation Especially in the Mathematical and Biomedical Sciences and Arts.* London: Gerald Duckworth & Company.

Davis, Martin. 2012. *The Universal Computer: The Road From Leibniz to Turing*. Boca Raton, London, New York: CRC Press.

De Vos, Ilse, Erika Gielen, Caroline Macé, and Peter Van Deun. 2010. "La Lettre B du Florilège Coislin: editio princeps." *Byzantion* 80: 72–120. https://www.jstor.org/stable/44173100.

DeRose, Steven J. 2004. "Markup Overlap: A Review and a Horse." In *Proceedings of Extreme Markup Languages®*. Montreal. http://conferences.idealliance.org/extreme/html/2004/DeRose01/EML2004DeRose01.html#t2.

DeRose, Steven J., David G. Durand, Elli Mylonas, and Allen H. Renear. 1990. "What Is Text, Really?" *Journal of Computing in Higher Education* 1 (2): 3–26.

Dijkstra, Edsger W. 1982. "On the Role of Scientific Thought (EWD447)." In *Selected Writings on Computing: A Personal Perspective*, 60–66. Texts and Monographs in Computer Science. New York, Heidelberg, Berlin: Springer-Verlag New York Inc.

Dinsmann, Melissa. 2016. "The Digital in the Humanities: An Interview with Franco Moretti." *Los Angeles Review of Books*, March 2, 2016. https://lareviewofbooks.org/article/the-digital-in-the-humanities-an-interview-with-franco-moretti/#!

Drucker, Johanna. 2003. "Designing Ivanhoe." *Text Technology* 12 (2): 19–41. http://texttechnology.mcmaster.ca/pdf/vol12_2_03.pdf.

———. 2010. "Graphesis: Visual Knowledge Production and Representation." *Poetess Archive Journal* 2 (1). https://journals.tdl.org/paj/index.php/paj/article/download/4/50.

———. 2011. "Humanities Approaches to Graphical Display." *Digital Humanities Quarterly* 5 (1). http://digitalhumanities.org/dhq/vol/5/1/000091/000091.html.

———. 2012. "Humanistic Theory and Digital Scholarship." In *Debates in Digital Humanities*, edited by Matthew K. Gold. Minneapolis: University of Minnesota Press. http://dhdebates.gc.cuny.edu/debates/text/34.

Edwards, Paul N. 1994. "Hyper Text and Hypertension: Post-Structuralist Critical Theory; Social Studies of Science, and Software." *Social Studies of Science* 24 (2): 229–278. http://pne.people.si.umich.edu/.

Efer, Thomas. 2017. "Graphdatenbanken für die textorientierten e-Humanities." Doctoral thesis, Leipzig: Universität Leipzig. http://nbn-resolving.de/urn:nbn:de:bsz:15-qucosa-219122.

Eide, Øyvind. 2014. "[Humanist] 28.360 Everything and Nothing." *Humanist Discussion Group*. http://lists.digitalhumanities.org/pipermail/humanist/2014-September/012314.html.

Fagin, Ronald, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. 1995. *Reasoning About Knowledge*. Cambridge: MIT Press.

Fiormonte, Domenico, and Cinzia Pusceddu. 2006. "The Text As a Product and As a Process. History, Genesis, Experiments." In *Manuscript, Variant, Genese - Genesis*, edited by Edward Vanhoutte and M. de Smedt, 109–128. Gent: KANTL. http://www.academia.edu/618689/The_Text_As_a_Product_and_As_a_Process._History_Genesis_Experiments.

Fischer, Franz. 2011. "About the HyperStack." Digital Edition (Hyperstack model). St. Patrick's Confessio. 2011. http://www.confessio.ie/about/hyperstack#.

Fish, Stanley. 2010. "The Crisis of the Humanities Officially Arrives,." *New York Times*, October 11, 2010, sec. The Opinion Pages. http://opinionator.blogs.nytimes.com/2010/10/11/the-crisis-of-the-humanities-officially-arrives.

———. 2011. "The Old Order Changeth." *New York Times*, December 26, 2011, sec. The Opinion Pages. http://opinionator.blogs.nytimes.com/2011/12/26/the-old-order-changeth/.

Flanders, Julia. 2009. "The Productive Unease of 21st Century Digital Scholarship." *DHQ: Digital Humanities Quarterly* 3 (3). http://www.digitalhumanities.org/dhq/vol/3/3/000055/000055.html.

———. 2012. "Modeling Scholarship." In *Knowledge Organization and Data Modeling in the Humanities*. Providence: Brown University. http://www.wwp.northeastern.edu/outreach/conference/kodm2012/flanders/flanders_paper.pdf.

Forbus, Kenneth D. 2008. "Qualitative Modeling." In *Handbook of Knowledge Representation*, edited by Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, 1:361–94. Foundations of Artificial Intelligence. Amsterdam, Boston, Heidelberg etc.: Elsevier.

Foucault, Michel. 1983. "Qu'est-ce qu'un auteur?" *(Conférence à la Société Française de Philosophie, le 22 février 1969, publiée dans le *Bulletin de la S.F.P.*, juillet-septembre 1969.)* *Littoral* 9: 3–32. http://sites.google.com/site/archivosdepablopeusner/Home/Qu%27est-cequ%27unauteur-Foucault.pdf?attredirects=0.

———. 1995. *Discipline and Punish: The Birth of the Prison.* Translated by Alan Sheridan. First published (France) 1975. New York: Vintage Books.

———. 1998. "What Is an Author?" In *Aesthetics, Method and Epistemology*, by Michel Foucault, edited by J. Faubion, translated by Robert Hurley, 2:187–204. Essential Works of Foucault 1954-1984. New York: The New Press.

Fowler, Martin. 2002. *Patterns of Enterprise Application Architecture*. Boston, San Francisco, etc.: Addison-Wesley Professional.

Frabetti, Federica. 2012. "Have the Humanities Always Been Digital?" In *Understanding Digital Humanities*, 161–171. London, New York: Palgrave Macmillan.

Froesse, Katrin. 2006. *Nietzsche, Heidegger, and Daoist Thought: Crossing Paths In-Between*. Albany: State University of New York Press.

Gabler, Hans Walter. 2010. "Theorizing the Digital Scholarly Edition." *Literature Compass* 7 (2): 43–56. https://doi.org/10.1111/j.1741-4113.2009.00675.x.

Galey, Alan, and Stan Ruecker. 2010. "How a Prototype Argues." *Literary and Linguistic Computing* 25 (4): 405–424. https://doi.org/10.1093/llc/fqq021.

Galison, Peter. 1999. "Objectivity Is Romantic." In *The Humanities and the Sciences*. Vol. 47. Occasional Paper. Philadelphia: American Council of Learned Societies. http://archives.acls.org/op/op47-3.htm#top.

———. 2010. "Trading with the Enemy." In *Trading Zones and Interactional Expertise: Creating New Kinds of Collaboration*, edited by Michael E. Gorman, 39–40. Cambridge, London: MIT Press.

Galloway, Alexander R. 2012. *The Interface Effect*. Cambridge: Polity Press.

Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, San Francisco, etc.: Addison-Wesley.

Gamut, L.T.F. 1991. *Logic, Language, and Meaning – Volume II: Intensional Logic and Logical Grammar*. Vol. 2. 2 vols. Chicago, London: The University of Chicago Press.

Gerbenzon, P. 1961. "Enkele opmerkingen over uitgavetechniek (naar aanleiding van: De Tweede Rose, uitgegeven door Prof. Dr. K. Heeroma)." *Tijdschrift voor Nederlandse Taal- en Letterkunde* 78: 20–31. http://www.dbnl.org/tekst/_tij003196101_01/_tij003196101_01_0003.php.

Gibbs, Fred, and Trevor Owens. 2013. "The Hermeneutics of Data and Historical Writing." In *Writing History in the Digital Age*, edited by Jack Dougherty and Kirsten Nawrotzki, 159–170. Ann Arbor: The University of Michigan Press. http://writinghistory.trincoll.edu/data/gibbs-owens-2012-spring.

Gitelman, Lisa, ed. 2013. *"Raw Data" Is an Oxymoron*. Cambridge: The MIT Press.

Goldfarb, Charles F. 1996. "The Roots of SGML – A Personal Recollection." 1996. http://www.sgmlsource.com/history/roots.htm.

———. 1997. "SGML: The Reason Why and the First Published Hint." *Journal of the American Society for Information Science* 48 (7): 656–661. https://doi.org/10.1002/(SICI)1097-4571(199707)48:7%3C656::AID-ASI13%3E3.0.CO;2-T.

Gorman, Michael E., Lekelia D. Jenkins, and Raina K. Plowright. 2012. "Human Interactions and Sustainability." In *Sustainability: Multi-Disciplinary Perspectives*, 88–111. Bentham Science Publishers.

Green, R.P.H., ed. 2008. *St. Augustine of Hippo, On Christian Teaching*. First published 1997, Reissued 1999. Oxford, New York: Oxford University Press.

Greetham, David. 1994. *Textual Scholarship: An Introduction*. Garland Reference Library of the Humanities 1417. New York & London: Garland Publishing Inc.

Greif, Irene, ed. 1988. *Computer-Supported Cooperative Work: A Book of Readings*. San Mateo: Morgan Kaufmann Publishers, Inc.

Grigar, Dene, Joseph Tabbi, Matthew Kirschenbaum, Michael Angelo Tata, Davin Heckman, Maria Angel, and Anna Gibbs. 2009. "E-Ject: On the Ephemeral Nature, Mechanisms, and Implications of Electronic Objects." In *Digital Arts and Culture 2009*. Irvine: University of California, Irvine. http://escholarship.org/uc/item/2xv6b6n0.

Hacking, Ian. 1981. "Do We See Through a Microscope?" *Pacific Philosophical Quarterly* 62 (4): 305–322.

Haentjens Dekker, Ronald, and David J. Birnbaum. 2017. "It's More than Just Overlap: Text As Graph." In *Proceedings of Balisage: The Markup Conference*

2017. Vol. 19. Balisage Series on Markup Technologies. Washington DC. https://doi.org/10.4242/BalisageVol19.Dekker01.

Haentjens Dekker, Ronald, Elli Bleeker, Bram Buitendijk, Astrid Kulsdom, and David J. Birnbaum. 2018. "TAGML: A Markup Language of Many Dimensions." In *Proceedings of Balisage: The Markup Conference 2018*. Balisage Series on Markup Technologies, Vol. 21. Washington D.C. https://doi.org/10.4242/BalisageVol21.HaentjensDekker01.

Haentjens Dekker, Ronald, Dirk Van Hulle, Gregor Middell, Vincent Neyt, and Joris Van Zundert. 2015. "Computer Supported Collation of Modern Manuscripts: CollateX and the Beckett Digital Manuscript Project." *Literary and Linguistic Computing* 30 (3): 452–470. https://doi.org/10.1093/llc/fqu007.

Haslhofer, Bernhard, Rainer Simon, Robert Sanderson, and Herbert Van de Sompel. 2011. "The Open Annotation Collaboration (OAC) Model." *Arxiv*. http://arxiv.org/abs/1106.5178.

Hawthorn, Jeremy. 2008. "Authority and the Death of the Author." In *Authority Matters: Rethinking the Theory and Practice of Authorship*, edited by Stephen Donovan, Danuta Fjellestad, and Rolf Lundén, 65–88. Amsterdam, New York: Rodopi.

Hayes-Sheen, Josh. 2017. "GraphViz Pocket Reference." November 12, 2017. https://graphs.grevian.org/.

Hayles, Katherine N. 2012. *How We Think: Digital Media and Contemporary Technogenesis*. Chicago, London: University of Chicago Press.

Heidegger, Martin. 2010. *Being and Time*. Translated by Joan Stambaugh. First published 1953, rev.with A foreword by Dennis J. Schmidt, Originally published in 1927 (German). Albany: State University of New York Press.

Hicks, Diana, Paul Wouters, Ludo Waltman, Sarah de Rijcke, and Ismael Rafols. 2015. "Bibliometrics: The Leiden Manifesto for Research Metrics." *Nature* 520 (April): 429–431. http://www.nature.com/news/bibliometrics-the-leiden-manifesto-for-research-metrics-1.17351.

Hiller, Moritz. 2015. "Signs o' the Times: The Software of Philology and a Philology of Software." *Digital Culture and Society* 1 (1): 152–163. https://doi.org/10.14361/dcs-2015-0110.

Hockey, Susan. 2004. "The History of Humanities Computing." In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth, 3–19. Malden, Oxford, Carlton: Blackwell Publishing. http://www.digitalhumanities.org/companion/.

Hoover, David L. 1999. *Language and Style in The Inheritors*. London, New York, Oxford: University Press Of America.

Hughes, Lorna H. 2004. *Digitizing Collections: Strategic Issues for the Information Manager*. Digital Futures Series. London: Facet Publishing.

Huitfeldt, Claus. 1995. "Multi-Dimensional Texts in a One-Dimensional Medium." *Computers and the Humanities* 28 (4/5): 235–241. http://www.jstor.org/stable/30204590.

Huitfeldt, Claus, and C.M. Sperberg-McQueen. 2008. "What Is Transcription?" *Literary and Linguistic Computing* 23 (3): 295–310. https://doi.org/10.1093/llc/fqn013.

Jannidis, Fotis, and Julia Flanders. 2013. "A Concept of Data Modeling for the Humanities." In *Digital Humanities 2013 Conference Abstracts*, 237–239. Lincoln: Center for Digital Research in the Humanities. http://dh2013.unl.edu/abstracts/ab-313.html.

Jansen, Leo, Hans Luijten, and Nienke Bakker, eds. 2009. *Vincent van Gogh: The Letters*. Amsterdam: Amsterdam University Press. http://www.vangoghletters.org/.

Jockers, Matthew L. 2013. *Macroanalysis: Digital Methods and Literary History*. Urabana, Chicago, Springfield: UI Press.

Jones, Steven E. 2014. *The Emergence of the Digital Humanities*. New York, London: Routledge.

Joyce, James. 1984. *Ulysses: A Critical and Synoptic Edition*. Edited by Hans Walter Gabler, Wolfhard Steppe, and Claus Melchior. New York, London: Garland Publishing Inc.

Karlsson, Lina, and Linda Malm. 2004. "Revolution or Remediation? A Study of Electronic Scholarly Editions on the Web." *HUMAN IT* 7 (1): 1–46. http://etjanst.hb.se/bhs/ith/1-7/lklm.pdf.

Kay, Alan C. 1993. "The Early History of Smalltalk." *ACM SIGPLAN Notices* 28 (3): 69–95. https://doi.org/10.1145/155360.155364.

Kegel, Peter. 2016. "Beyond the Great Divide: New Explorations in Editing the Complete Works of Dutch Novelist Willem Frederik Hermans." *Digital Studies / Le Champ Numérique* 6 (2): (special issue) *(2015-2016) Beyond Accessibility: Textual Studies in the Twenty-First Century*, edited by Brent Nelson and Richard Cunningham. https://doi.org/10.16995/dscn.5.

Kestemont, Mike. 2012. "Het gewicht van de auteur. Een onderzoek naar stylometrische auteursherkenning in de Middelnederlandse epiek." Antwerpen: Universiteit Antwerpen, Faculteit Letteren en Wijsbegeerte, Departementen Taal- en Letterkunde.

Kirsch, Adam. 2014. "Technology Is Taking Over English Departments: The False

Promise of the Digital Humanities." *New Republic*, May 2, 2014. http://www.newrepublic.com/article/117428/limits-digital-humanities-adam-kirsch.

Kirschenbaum, Matthew. 2008. *Mechanisms: New Media and the Forensic Imagination*. Cambridge, London: The MIT Press.

———. 2012. "What Is Digital Humanities and What's It Doing in English Departments?" In *Debates in the Digital Humanities*, edited by Matthew K. Gold, 3–11. Minneapolis: University of Minnesota Press. http://dhdebates.gc.cuny.edu/debates/text/38.

Kirschenbaum, Matthew G., Erika L. Farr, Kari M. Kraus, Naomi Nelson, Catherine Stollar Peters, Gabriela Redwine, and Doug Reside. 2009. "Digital Materiality: Preserving Access to Computers as Complete Environments." In *IPRES 2009: The Sixth International Conference on the Preservation of Digital Objects*. San Francisco: California Digital Library. http://escholarship.org/uc/item/7d3465vg.

Kittler, Friedrich. 1993a. *Draculas Vermächtnis*. Leipzig: Reclam Verlag. http://monoskop.org/images/b/bo/Kittler_Friedrich_Draculas_Vermaechtnis_Technische_Schriften.pdf.

———. 1993b. "Es gibt keine Software." In *Draculas Vermächtmis*, 225–242. Leipzig: Reclam Verlag.

Klein, Hans K., and Daniel Lee Kleinman. 2002. "The Social Construction of Technology: Structural Considerations." *Science, Technology, & Human Values* 27 (1): 28–52. https://doi.org/10.1177/016224390202700102.

Knuth, Donald E. 1984. "Literate Programming." *The Computer Journal* 27 (1): 97–111. https://doi.org/10.1093/comjnl/27.2.97.

Kouw, Matthijs, Charles Van den Heuvel, and Andrea Scharnhorst. 2013. "Exploring Uncertainty in Knowledge Representations: Classifications, Simulations, and Models of the World." In *Virtual Knowledge: Experimenting in the Humanities and the Social Sciences*, edited by Paul Wouters, Anne Beaulieu, Andrea Scharnhorst, and Sally Wyatt, 89–125. Cambridge: MIT Press.

Kramer, Adam D.I., Jamie E. Guillory, and Jeffrey T. Hancock. 2014. "Experimental Evidence of Massive-Scale Emotional Contagion through Social Networks." *Proceedings of the National Academy of Sciences of the United States of America* 111 (24): 8788–8790. https://doi.org/10.1073/pnas.1320040111.

Kuhn, Thomas S. 2012. *The Structure of Scientific Revolutions*. 4 (first published 1962). New York and London: The University of Chicago Press.

Kwa, Chunglin. 2011. *Styles of Knowing: A New History of Science from Ancient Times to the Present*. Pittsburgh: University of Pittsburgh Press.

Landow, George P. 1994. *Hyper/Text/Theory*. Baltimore: Johns Hopkins University Press.

———. 2006. *Hypertext 3.0: Critical Theory and New Media in an Era of Globalization.* Rev. ed. of Hypertext 2.0 1997. Parallax Revisions of Culture and Society. Baltimore: The John Hopkins University Press.

Laplante, Phillip A. 2007. *What Every Engineer Should Know About Software Engineering.* What Every Engineer Should Know 40. Boca Raton: Taylor & Francis Group, LLC.

Latour, Bruno. 1987. *Science in Action: How to Follow Scientists and Engineers Through Society.* Eleventh printing, 2003. Cambridge: Harvard University Press.

Latour, Bruno, and Steve Woolgar. 1986. *Laboratory Life: The Construction of Scientific Facts.* Princeton: Princeton University Press.

Le Roy Ladurie, Emmanuel. 2014. "L'historien et l'ordinateur." In *Le territoire de l'historien*, reprint (ePub), first edition 1973, originally appeared in Le Nouvel Observateur, 8 mai 1968, 12–14. Bibliothèque des Histoires. Paris: Gallimard.

Lee, G., and W. Xia. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility." *MIS Quarterly* 34: 87–114.

Lehman, M.M., and J.F. Ramil. 2000. "Software Evolution in the Age of Component-Based Software Engineering." *Software, IEE Proceedings* 147 (6): 249–255. https://doi.org/10.1049/ip-sen:20000922.

Lehrer, Jonah. 2011. "Steve Jobs: 'Technology Alone Is Not Enough.' " *New Yorker*, October 7, 2011. https://www.newyorker.com/news/news-desk/steve-jobs-technology-alone-is-not-enough.

Leonardi, Paul M. 2011. "When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies." *MIS Quarterly* 35 (1): 147–167.

Lernout, Geert. 2006. "Controversial Editions: Hans Walter Gabler's Ulysses." *Text* 16: 229–241.

Levallois, Clement, Stephanie Steinmetz, and Paul Wouters. 2013. "Sloppy Data Floods or Precise Methodologies? Dilemmas in the Transition to Data-Intensive Research in Sociology and Economics." In *Virtual Knowledge: Experimenting in the Humanities and the Social Sciences*, 151–182. Cambridge: MIT Press.

Lin, Yi, Mian-yun Chen, and Sifeng Liu. 2004. "Theory of Grey Systems: Capturing Uncertainties of Grey Information." *Kybernetics* 33 (2): 196–218. https://doi.org/10.1108/03684920410514139.

Love, Harold. 2002. *Attributing Authorship: An Introduction*. Cambridge, New York, etc.: Cambridge University Press.

Lunenfeld, Peter, Anne Burdick, Johanna Drucker, Todd Presner, and Jeffrey Schnapp. 2012. *Digital_Humanities*. Cambridge, London: MIT Press. http://mitpress.mit.edu/books/digitalhumanities-0.

Mallery, John C., Roger Hurwitz, and Gavan Duffy. 1986. "Hermeneutics: From Textual Explanation to Computer Understanding?" Memo AIM-871. AI Memos. MIT. http://hdl.handle.net/1721.1/6438.

Mambrol, Nasrullah. 2016. "Roland Barthes' Concept of Readerly and Writerly Texts." *Literary Theory and Criticism*. March 21, 2016. https://literariness.org/2016/03/21/roland-barthes-concept-of-readerly-and-writerly-texts/.

Manovich, Lev. 2013. *Software Takes Command*. Vol. 5. International Texts in Critical Media Aesthetics. New York, London, New Delhi etc.: Bloomsbury Academic.

Marchesi, Simone. 2011. *Dante and Augustine: Linguistics, Poetics, Hermeneutics*. Toronto: University of Toronto Press.

Marino, Mark C. 2006. "Critical Code Studies." *Electronic Book Review*, December. http://www.electronicbookreview.com/thread/electropoetics/codology.

———. 2014. "Field Report for Critical Code Studies, 2014." *Computational Culture—A Journal of Software Studies*, no. 4 (November). http://computationalculture.net/article/field-report-for-critical-code-studies-2014%E2%80%A8.

Marinova, Dora, and John Phillimore. 2003. "Models of Innovation." In *The International Handbook on Innovation*, edited by Larisa V. Shavinina, 44–53. Kidlington, Oxford: Elsevier Science Ltd.

Martin, Robert C. 2003. *Agile Software Development: Principles, Patterns, and Practices*. Alan Apt Series. Upper Saddle River, New Jersey: Prentice Hall.

Mastop, Rosja. 2011. "Modal Logic for Artificial Intelligence." Semantic Scholar. https://www.semanticscholar.org/paper/Modal-Logic-for-Artificial-Intelligence-Mastop/ec61f12c65a7e704d9b44d6cb46cd7417029bfe1#references.

Mathijsen, Marita. 2003. *Naar de letter: Handboek editiewetenschap*. 3rd (unchanged) edition, first published 1995. Den Haag: Constantijn Huygens Instituut van de Koninklijke Nederlandse Akademie van Wetenschappen. https://www.dbnl.org/tekst/math004naar01_01/.

McCarty, Willard. 2005. *Humanities Computing*. Basingstoke, New York: Palgrave Macmillan.

———. 2013. "[Humanist] 27.124 Computationalists and Humanists." *Humanist Discussion Group*. http://lists.digitalhumanities.org/pipermail/humanist/201

3-June/011052.html.

———. 2014. "Getting There from Here. Remembering the Future of Digital Humanities: Roberto Busa Award Lecture 2013." *Literary and Linguistic Computing* 29 (3): 283–306. https://doi.org/10.1093/llc/fqu022.

McGann, Jerome. 1991. *The Textual Condition*. Princeton: Princeton University Press.

———. 1995. "The Rationale of Hypertext." The Rationale of Hypertext. May 6, 1995. http://www2.iath.virginia.edu/public/jjm2f/rationale.html.

———. 2001. *Radiant Textuality: Literature after the World Wide Web*. New York: Palgrave Macmillan.

———. 2003. "Texts in N-Dimensions and Interpretation in a New Key." *Text Technology* 12 (2): 1–18. http://texttechnology.mcmaster.ca/pdf/vol12_2_02.pdf.

———. 2004. "Marking Texts of Many Dimensions." In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth. Oxford: Blackwell. http://www.digitalhumanities.org/companion/view?docId=blackwell/9781405103213/9781405103213.xml&chunk.id=ss1-3-4&toc.depth=1&toc.id=ss1-3-4&brand=default.

———. 2010. "Electronic Archives and Critical Editing." *Literature Compass* 7: 37–42. https://doi.org/10.1111/j.1741-4113.2009.00674.x.

———. 2013. "Philology in a New Key." *Critical Inquiry* 39 (2): 327–346. http://www.jstor.org/stable/10.1086/668528.

———. 2015. "Truth and Method: Humanities Scholarship as a Science of Exceptions." *Interdisciplinary Science Review* 40 (2): 204–218.

McKenzie, D.F. 1999. *Bibliography and the Sociology of Texts*. Cambrdige: Cambridge University Press.

McPherson, Tara. 2012. "Why Are the Digital Humanities So White? Or Thinking the Histories of Race and Computation." In *Debates in the Digital Humanities*, edited by Matthew K. Gold, 139–60. Minneapolis: University of Minnesota Press. http://dhdebates.gc.cuny.edu/debates/text/29.

Meister, Jan Christoph. 1995. "Consensus Ex Machina? Consensus qua Machina!" *Literary and Linguistic Computing* 10 (4): 263–270. https://doi.org/10.1093/llc/10.4.263.

———. 2013. "[Humanist] 27.125 Computationalists and Humanists." *Humanist Discussion Group*. http://lists.digitalhumanities.org/pipermail/humanist/2013-June/011053.html.

Millen, David, Sally Wyatt, Lora Aroyo, Rens Bod, Antal Van den Bosch, Irene Greif, Antske Fokkens, et al. 2013. "Meaning and Perspectives in the Digital Humanities: A White Paper for the Establishment of a Center for Humanities and Technology (CHAT).” Edited by David Millen and Sally Wyatt. Royal Netherlands Academy of Arts and Sciences, University of Amsterdam, VU University Amsterdam, Netherlands eScience Center, IBM.

Mitra, Ananda. 1999. “Characteristics of the WWW Text: Tracing Discursive Strategies.” *Journal of Computer-Mediated Communication* 5 (1). https://doi.org/10.1111/j.1083-6101.1999.tb00330.x.

Montfort, Nick, Patsy Baudoin, John Bell, Ian Bogost, Jeremy Douglass, Mark Marino, Michael Mateas, Casey Reas, Mark Sample, and Noah Vawter. 2012. *10 PRINT CHR$(205.5+RND(1));: GOTO 10*. Cambridge, London: The MIT Press. http://10print.org/.

Moretti, Franco. 2007. *Graphs, Maps, Trees: Abstract Models for Literary History*. London: Verso.

———. 2013. *Distant Reading*. London: Verso.

Morozov, Evgeny. 2013. *To Save Everything, Click Here: The Folly of Technological Solutionism*. New York: PublicAffairs.

Murphy, J. Stephen. 2008. “The Death of the Editor.” *Essays in Criticism* 58 (4): 289–310. https://doi.org/10.1093/escrit/cgn020.

Myers, Rob. 2012. “Human Readble Messages.” *Furtherfield*, June 4, 2012. http://www.furtherfield.org/features/reviews/human-readable-messages.

Natoli, Joseph, and Linda Hutcheon, eds. 1993. *A Postmodern Reader*. Albany: State University of New York Press.

Neill, Iian, and Andreas Kuczera. 2019. “The Codex – an Atlas of Relations.” *Zeitschrift Für Digitale Geisteswissenschaften* Sonderbände 4 (Die Modellierung des Zweifels – Schlüsselideen und-konzepte zur graphbasierten Modellierung von Unsicherheiten). https://doi.org/10.17175/sb004_008.

Nelson, Theodor Holm. 1993. *Literary Machines. The Report on, and of, Project Xanadu Concerning Word Processing, Electronic Publishing, Hypertext, Thinkertoys, Tomorrow’s Intellectual Revolution, and Certain Other Topics Including Knowledge, Education and Freedom.* First published 1981. Mindfull Press.

———. 1995. “The Heart of Connection: Hypermedia Unified by Transclusion.” *Communications of the ACM* 38 (8): 31–33. https://doi.org/10.1145/208344.208353.

———. 2010. *POSSIPLEX: Movies, Intellect, Creative Control, My Computer Life and the Fight for Civilization*.

Nickles, Thomas. 2003. “Evolutionary Models of Innovation and the Meno Problem.” In *The International Handbook on Innovation*, edited by Larisa V. Shavinina, 54–78. Amsterdam, Boston, Heidelberg etc.: Elsevier Ltd.

Nowviskie, Bethany. 2012. "Too Small to Fail." *Bethany Nowviskie*. October 13, 2012. http://nowviskie.org/2012/too-small-to-fail/.

Nyhan, Julianne, and Oliver Duke–Williams. 2014. "Joint and Multi-Authored Publication Patterns in the Digital Humanities." *Literary & Linguistic Computing* 29 (4). https://doi.org/10.1093/llc/fqu018.

Opas-Hänninen, Lisa Lena. 1995. "Special Section: New Approaches to Computer Applications in Literary Studies." *Literary and Linguistic Computing* 10 (4): 261–262. https://doi.org/10.1093/llc/10.4.261.

Parry, Milman. 1971. "The Historical Method in Literary Criticism." In *The Making of Homeric Verse: The Collected Papers of Milman Parry*, edited by Adam Parry, Reprint of Parry, Milman. 1936. The historical method in literary criticism. Harvard Alumni Bulletin 38 (1936): 778–82., 408–13. Oxford: Clarendon Press.

Pierazzo, Elena. 2011. "A Rationale of Digital Documentary Editions." *Literary and Linguistic Computing* 26 (4): 463–477.

Piersma, Hinke, and Kees Ribbens. 2013. "Digital Historical Research: Context, Concepts and the Need for Reflection." *Low Countries Historical Review* 128 (4): 78–102. http://www.bmgn-lchr.nl/index.php/bmgn/article/view/URN%3ANBN%3ANL%3AUI%3A10-1-110023.

Piper, Andrew. 2015. "Novel Devotions: Conversional Reading, Computational Modeling, and the Modern Novel." *New Literary History* 46 (1): 63–98. https://doi.org/10.1353/nlh.2015.0008.

Porter, Dot. 2013. "Medievalists and the Scholarly Digital Edition." *Scholarly Editing: The Annual of the Association for Documentary Editing* 34. http://www.scholarlyediting.org/2013/essays/essay.porter.html.

Posy, Carl J. 2013. "Computability and Constructibility." In *Computability: Turing, Gödel, Curch, and Beyond*, edited by B. Jack Copeland, Carl J. Posy, and Oron Shagrir, 105–140. Cambridge, London: The MIT Press.

Pratt, Vaughan R. 1976. "Semantical Considerations on Floyd-Hoare Logic." In *Proceedings of the 17th Annual IEEE Symposium on the Foundations of Computer Science*, 109–121.

———. 1980. "Application of Modal Logic to Programming." *Studia Logica: An International Journal for Symbolic Logic* 39 (2/3): 257–274. http://www.jstor.org/stable/20014985.

Price, Kenneth M. 2004. "Electronic Scholarly Editions." In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth. Oxford: Blackwell. http://www.digitalhumanities.org/companion/view

?docId=blackwell/9781405148641/9781405148641.xml&chunk.id=ss1-6-5&toc.depth=1&toc.id=ss1-6-5&brand=9781405148641_brand.

"Published." n.d. eLaborate. Accessed April 2, 2019. http://elaborate.huygens.knaw.nl/?page_id=133.

Raley, Rita. 2002. "Interferences: [Net.Writing] and the Practice of Codework." *Electronic Book Review*, September. http://www.electronicbookreview.com/thread/electropoetics/net.writing.

Ramberg, Bjørn, and Kristin Gjesdal. 2013. "Hermeneutics." The Stanford Encyclopedia of Philosophy. http://plato.stanford.edu/archives/sum2013/entries/hermeneutics/.

Ramsay, Stephen. 2010. "The Hermeneutics of Screwing Around; or What You Do with a Million Books." http://www.playingwithhistory.com/wp-content/uploads/2010/04/hermeneutics.pdf.

———. 2011a. "Who's In and Who's Out." *Stephen Ramsay — Blog*. January 8, 2011. https://web.archive.org/web/20170721063833/http://stephenramsay.us/text/2011/01/08/whos-in-and-whos-out/.

———. 2011b. "On Building." *Stephen Ramsay — Blog* January 11, 2011. http://stephenramsay.us/text/2011/01/11/on-building/.

———. 2011c. *Reading Machines: Toward an Algorithmic Criticism (Topics in the Digital Humanities)*. Chicago: University of Illinois Press.

Ravasio, P., and V. Tscherter. 2004. "The Users Theories on the Desktop Metaphor – or Why We Should Seek Metaphor-Free Interfaces." In *Beyond the Desktop Metaphor: Designing Integrated Digital Work Environments*, edited by Victor Kaptelinin and M. Czerwinski, 265–294. Cambridge: MIT Press.

Rawson, Katie, and Trevor Muñoz. 2016. "Against Cleaning." *Curating Menus* July 6, 2016. http://curatingmenus.org/articles/against-cleaning/.

Rayward, W. Boyd. 1994. "Visions of Xanadu: Paul Otlet (1868-1944) and Hypertext." *JASIS* 45: 235–250. http://people.lis.illinois.edu/~wrayward/otlet/xanadu.htm.

Reenen, van, Pieter, August Hollander, den, and Margot Mulken, van, eds. 2004. *Studies in Stemmatology II*. Amsterdam, Philadelphia: John Benjamins Publishing Company.

Renear, Allen. 2004. "Text Encoding." In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth, 218–239. Malden, Oxford, Carlton: Blackwell Publishing. http://www.digitalhumanities.org/companion/view?docId=blackwell/9781405103213/9781405103213.xml&chunk.id=ss1-3-5&toc.depth=1&toc.id=ss1-3-5&brand=default.

Riddell, Allen, and Karina Van Dalen-Oskam. 2018. "Readers and Their Roles:

Evidence from Readers of Contemporary Fiction in the Netherlands." *PLOS ONE* 13 (7): e0201157. https://doi.org/10.1371/journal.pone.0201157.

Ries, Thorsten. 2010. "„die geräte klüger als ihre besitzer": Philologische Durchblicke hinter die Schreibszene des Graphical User Interface: Überlegungen zur digitalen Quellenphilologie, mit einer textgenetischen Studie zu Michael Speiers ausfahrt st. nazaire." *Edito* 24 (December): 149–199. https://doi.org/10.1515/9783110223163.0.149.

Ritchie, D.M. 1984. "The UNIX System: The Evolution of the UNIX Time-Sharing System." *AT&T Bell Laboratories Technical Journal* 63 (8): 1577–1593. https://doi.org/10.1002/j.1538-7305.1984.tb00054.x.

Robinson, Peter. 1989. "The Collation and Textual Criticism of Icelandic Manuscripts (1): Collation." *Literary and Linguistic Computing* 4 (2): 99–105. https://doi.org/10.1093/llc/4.2.99.

———. 2004. "Where We Are with Electronic Scholarly Editions, and Where We Want to Be." March 24, 2004. http://computerphilologie.uni-muenchen.de/jg03/robinson.html.

———. 2013a. "Towards a Theory of Digital Editions." *Variants* 10: 105–131. https://brill.com/view/book/edcoll/9789401209021/B9789401209021-s009.xml.

———. 2013b. "Five Desiderata for Scholarly Editions in Digital Form." In *Digital Humanities 2013 Conference Abstracts*, 355–356. Lincoln: Center for Digital Research in the Humanities. http://dh2013.unl.edu/abstracts/ab-314.html.

———. 2013c. "Why Digital Humanists Should Get out of Textual Scholarship. And If They Don't, Why We Textual Scholars Should Throw Them Out." *Scholarly Digital Editions*. July 29, 2013. http://scholarlydigitaleditions.blogspot.nl/2013/07/why-digital-humanists-should-get-out-of.html.

———. 2016. "The Digital Revolution in Scholarly Editing." In *Ars Edendi Lecture Series, Vol. IV*, edited by Barbara Crostini, Gunilla Iversen, and Brian Jensen, 181–207. Stockholm: Stockholm University Press. https://doi.org/10.16993/baj.

Rockwell, Geoffrey. 2003. "What Is Text Analysis, Really?" *Literary and Linguistic Computing* 18 (2): 209–219.

Roelli, Philipp. 2015. "Copy Text." In *Parvum Lexicon Stemmatologicum*. https://wiki.helsinki.fi/display/stemmatology/Copy+text.

Roelli, Philipp, and Dieter Bachmann. 2010. "Towards Generating a Stemma of Complicated Manuscript Traditions: Petrus Alfonsi's Dialogus." *Revue d'histoire Des Textes* 5 (2010): 307–321. https://doi.org/10.1484/J.RHT.5.101260.

Rogers, Everett M. 1983. *Diffusion of Innovations*. Third edition. New York, London: The Free Press.

Rorty, Richard. 1988. "Taking Philosophy Seriously." *New Republic* 198 (15): 31–34.

Russell, Stuart, and Peter Norvig. 2009. *Artificial Intelligence: A Modern Approach*. 3rd edition. Prentice Hall.

Sahle, Patrick. 2013. *Digitale Editionsformen, Zum Umgang mit der Überlieferung unter den Bedingungen des Medienwandels. Teil 2: Befunde, Theorie und Methodik.* [Finale Print-Fassung]. Vol. 8. Schriften des Instituts für Dokumentologie und Editorik. Norderstedt: Norderstedt Books on Demand. https://kups.ub.uni-koeln.de/5352/.

Schmidt, Benjamin M. 2016. "Do Digital Humanists Need to Understand Algorithms?" In *Debates in the Digital Humanities 2016*, edited by Matthew K. Gold and Lauren F. Klein. Minneapolis: University of Minnesota Press. http://dhdebates.gc.cuny.edu/debates/text/99.

Schmidt, D., and R. Colomb. 2009. "A Data Structure for Representing Multi-Version Texts Online." *International Journal of Human-Computer Studies* 67 (6): 497–514.

Schmidt, Desmond. 2014. "The Slow Tide Turns…" *Multi-Version Documents*. March 14, 2014. http://multiversiondocs.blogspot.nl/2014/03/the-slow-tide-turns.html.

Schreibman, Susan, Laura Mandell, and Stephen Olsen. 2011. "Introduction." *Profession*, no. 6: 123–201. https://doi.org/10.1632/prof.2011.2011.1.123.

Scott, Marc. 2013. "Kids Can't Use Computers… And This Is Why It Should Worry You." *Coding 2 Learn: Education and Technology Ramblings with a Little Politics for Good Measure*. July 29, 2013. http://coding2learn.org/blog/2013/07/29/kids-cant-use-computers/.

Sculley, D., and Bradley M. Pasanek. 2008. "Meaning and Mining: The Impact of Implicit Assumptions in Data Mining for the Humanities." *Literary and Linguistic Computing* 23 (4): 409–424. http://llc.oxfordjournals.org/content/23/4/409.abstract.

Seaver, Nick. 2013. "Knowing Algorithms." In *Media in Transition 8*. Cambridge: MIT. http://nickseaver.net/s/seaverMiT8.pdf.

Shannon, C.E. 1948. "A Mathematical Theory of Communication." *The Bell System Technical Journal* 27 (3): 379–423.

Shaw, Ryan, Michael Buckland, and Patrick Golden. 2013. "Open Notebook Humanities: Promise and Problems." In *Digital Humanities 2013 Conference Abstracts*, 389–392. Lincoln: University of Nebraska–Lincoln. http://dh2013.unl.edu/abstracts/ab-297.html.

Shillingsburg, Peter. 2013. "Is Reliable Social Scholarly Editing an Oxymoron?" In *Social, Digital, Scholarly Editing*. Saskatoon: University of Saskatchewan. http://ecommons.luc.edu/ctsdh_pubs/1.

Shillingsburg, Peter L. 2006. *From Gutenberg to Google: Electronic Representations of Literary Texts*. Cambridge, New York, Melbourne, etc.: Cambridge University Press.

Shirky, Clay. 2010. *Cognitive Surplus: Creativity and Generosity in a Connected Age*. New York: Pinguin Group.

Siemens, Ray, Meagan Timney, Cara Leitch, Corina Koolen, and Alex Garnett. 2012. "Toward Modeling the Social Edition: An Approach to Understanding the Electronic Scholarly Edition in the Context of New and Emerging Social Media." *Literary and Linguistic Computing* 27 (4): 445–461. https://doi.org/10.1093/llc/fqs013.

Simon, Herbert A. 1995. "Technology Is Not the Problem." In *Speaking Minds: Interviews with Twenty Eminent Cognitive Scientists*, edited by P. Baumgarter and S. Payr, 232–248. Princeton University Press.

Small, Helen. 2013. *The Value of the Humanities*. Oxford: Oxford University Press.

Smiraglia, Richard P. 2001. *The Nature of "a Work": Implications for the Organization of Knowledge.* Maryland: Scarecrow Press.

Smith, James. 2012. "Coding and Digital Humanities." *James Gottlied – Seeing What Happens When You Collide the Humanities with the Digital.* March 8, 2012. http://www.jamesgottlieb.com/old/2012/03/coding-and-digital-humanities/.

Snow, C.P. 1998. *The Two Cultures*. Part 1 first published 1959. Cambridge, New York, etc.: Cambridge University Press.

Sperberg-McQueen, C.M. 1994. "Textual Criticism and the Text Encoding Initiative." http://www.tei-c.org/Vault/XX/mla94.html.

———. 2002. "What Matters?" In *Proceedings of Extreme Markup Languages*. Montréal. http://conferences.idealliance.org/extreme/html/2002/CMSMcQ02/EML2002CMSMcQ02.html.

Szondi, Peter, and Timothy Bahtl. 1978. "Introduction to Literary Hermeneutics." *New Literary History* 10 (1): 17–29. http://www.jstor.org/stable/468303.

Teeuwen, Mariken. 2011. "Writing between the Lines: Reflections of Scholarly Debate in a Carolingian Commentary Tradition." In *Carolingian Scholarship and Martianus Capella: Ninth-Century Commentary Traditions on "De Nuptiis" in Context*, edited by Mariken Teeuwen and Sinaed O'Sullivan, 11–34. Cultural Encounters in Late Antiquity and the Middle Ages (CELAMA) 12. Turnhout:

Brepols. https://www.brepolsonline.net/doi/abs/10.1484/M.CELAMA-EB.4.300.

Teeuwen, Mariken, S. O'Sullivan, I. Savelskous, M. Garrison, Th. Brouwer, A. Sroczynski, A.P. Orbán, N. Lozovsky, B.S. Eastwoord, and J.Y. Guillaumin, eds. 2008. *Carolingian Scholarship and Martianus Capella: The Oldest Commentary Tradition*. First edition. Den Haag: Huygens Institute. http://martianus.huygens.knaw.nl/.

Terras, M., E. Priego, A. Liu, G. Rockwell, S. Sinclair, C. Hensler, and L. Thomas. 2013. " 'The Humanities Matter!' Infographic." http://4humanities.org/infographic.

Terras, Melissa, Julianne Nyhan, and Edward Vanhoutte, eds. 2013. *Defining Digital Humanities*. Farnham, Burlington: Ashgate.

Thaller, Manfred. 2018. "On Information in Historical Sources." *A Digital Ivory Tower*. April 24, 2018. https://ivorytower.hypotheses.org/56#more-56.

Thiruvathukal, George, Steven Jones, and Peter Shillingsburg. 2010. "E-Carrel: An Environment for Collaborative Textual Scholarship." *Journal of the Chicago Colloquium on Digital Humanities and Computer Science* 1 (2): 1–12. https://ecommons.luc.edu/cs_facpubs/1/.

Thuan, Duc. 2000. "D a Y s o F J a v A m O o N (Fragment)." 2000. https://web.archive.org/web/20050114091845/http://www.ducthuan.com/JavaMoon/moon11.htm.

Timberg, Scott. 2013. "David Lowery: Silicon Valley Must Be Stopped, or Creativity Will Be Destroyed - Salon.Com." Newspaper. SALON. December 4, 2013. http://www.salon.com/2013/12/04/david_lowery_silicon_valley_must_be_stopped_or_creativity_will_be_destroyed/.

Timpanaro, Sebastiano. 2005. *The Genesis of Lachmann's Method*. Edited and Translated by Glenn W. Most. Chicago, London: The University of Chicago Press.

Trovato, Paolo. 2014. *Everything You Always Wanted to Know about Lachmann's Method: A Non-Standard Handbook of Genealogical Textual Criticism in the Age of Post-Structuralism, Cladistics, and Copy-Text*. Translated by Federico Poole. Padova: Libreriauniversitaria.it Edizioni.

Underwood, Ted. 2019. *Distant Horizons: Digital Evidence and Literary Change*. Chicago and London: The University of Chicago Press.

Underwood, Ted, and Jordan Sellers. 2016. "The *Longue Durée* of Literary Prestige." *Modern Language Quarterly* 77 (3): 321–344. https://doi.org/10.1215/00267929-3570634.

Unsworth, John. 2000. "Scholarly Primitives: What Methods Do Humanities Researchers Have in Common, and How Might Our Tools Reflect This?" In *Humanities Computing: formal methods, experimental practice*. London: King's College. http://www.people.virginia.edu/~jmu2m/Kings.5-00/primitives.html.

———. 2002. "What Is Humanities Computing And What Is Not?" Edited by Georg Braungart, Peter Gendolla, and Fotis Jannidis. *Jahrbuch Für Computerphilologie* 4. http://computerphilologie.digital-humanities.de/jg02/unsworth.html.

Van Dalen-Oskam, Karina, D. 2014. "Mocht Je Even Tijd Hebben: Ordening by 'relevance' in ELaborate." Email to author. January 10, 2014.

Van Dalen-Oskam, Karina, and Joris J. Van Zundert. 2007. "Delta for Middle Dutch: Author and Copyist Distinction in *Walewein*." *Literary and Linguistic Computing* 22 (3): 345–362. https://doi.org/10.1093/llc/fqm012.

Van den Heuvel, C. 2012. "Circulation of Knowledge in the Digital Republic of Letters. Making Correspondences of Manuscripts and Printed Editions Accessible for Research." presented at the 5th Liber Manuscript Conference Promoting access to manuscript content, Paris Bibliothèque Nationale de France, May 29. http://ckcc.huygens.knaw.nl/wp-content/bestanden/2012/05/CHeuvel_LIBER_ParijsDEF.pdf.

Van den Heuvel, C., and W. Boyd Rayward. 2011. "Facing Interfaces: Paul Otlet's Visualizations of Data Integration." *Journal of the American Society for Information Science and Technology* 62 (12): 2313–2326.

Van Hulle, Dirk. 2004. *Textual Awareness: A Genetic Study of Late Manuscripts by Joyce, Proust, and Mann*. University of Michigan Press.

———. 2010. "Editing Samuel Beckett." Workshop lecture presented at the Interedition Workshop National Library of Israel, Jerusalem, October 6.

———. 2015. "Editorial Principles and Practice." Digital Edition (Archive). Samuel Beckett: Digital Manuscript Project. November 2015. http://www.beckettarchive.org/editorial.jsp.

Van Hulle, Dirk, and Mark Nixon. 2011. "Samuel Beckett Digital Manuscript Project." Scholarly edition/archive. Samuel Beckett Digital Manuscript Project. 2011. http://www.beckettarchive.org/.

Van Mierlo, Wim. 2007. "Textual Scholarship and the Material Book." *Variants: The Journal of the European Society for Textual Scholarship* 6. http://www.academia.edu/209627/Introduction_to_Textual_Scholarship_and_the_Material_Book.

Van Reenen, Pieter, and Margot Mulken, eds. 1996. *Studies in Stemmatology*. With the assistance of Janet Dyk. Amsterdam, Philadelphia: John Benjamins Pub.

Co. http://benjamins.com/#catalog/books/z.79/toc.

Van Zundert, Joris J. 2012. "If You Build It, Will We Come? Large Scale Digital Infrastructures as a Dead End for Digital Humanities." *Historical Social Research–Historische Sozialforschung* 37 (3): 165–186. http://www.youtube.com/watch?v=LZumq17LDaI&feature=youtube_gdata_player.

———. 2018. "Why We Should Think About a Domain Specific Computer Language (DSL) for Scholarship." In *Digital Hermeneutics in History: Theory and Practice*. Esch-sur-Alzette: C2DH of Luxembourg University. http://jorisvanzundert.net/blogposts/why-we-should-think-about-a-domain-specific-computer-language-dsl-for-scholarship/.

Van Zundert, Joris J., Smiljana Antonijevic, Karina Van Dalen-Oskam, Anne Beaulieu, Douwe Zeldenrust, and Tara Andrews. 2012. "Cultures of Formalization: Towards an Encounter Between Humanities and Computing." In *Understanding Digital Humanities: The Computational Turn and New Technology*, edited by David Berry, 279–294. London: Palgrave Macmillan.

Van Zundert, Joris, Douwe Zeldenrust, and Anne Beaulieu. 2009. "Alfalab: Construction and Deconstruction of a Digital Humanities Experiment." In *Proceedings: 2009 Fifth IEEE International Conference on e-Science e-Science 2009*, 1–5. Oxford: IEEE Computer Society. https://doi.org/10.1109/e-Science.2009.8.

Vandoulakis, Ioannis M. 2015. "On A.A. Markov's Attitude towards Brouwer's Intuitionism." *Philosophia Scientiæ* 19 (1): 143–158. https://journals.openedition.org/philosophiascientiae/1054.

Vanhoutte, Edward. 2009. "Paul Otlet (1868-1944) and Vannevar Bush (1890-1974)." *The Mind Tool: Edward Vanhoutte's Blog*. March 1, 2009. http://edwardvanhoutte.blogspot.nl/2009/03/paul-otlet-1868-1944-and-vannevar-bush.html.

———. 2011. "So You Think You Can Edit? The Masterchef Edition." *The Mind Tool: Edward Vanhoutte's Blog*. October 14, 2011. http://edwardvanhoutte.blogspot.nl/2011/10/so-you-think-you-can-edit-masterchef.html.

Vee, Annette. 2013. "Understanding Computer Programming as a Literacy." *LiCS* 1 (2): 42–64. http://licsjournal.org/OJS/index.php/LiCS/article/view/24/26.

Vitali, Fabio. 2016. "The Expressive Power of Digital Formats: Criticizing the Manicure of the Wise Man Pointing at the Moon." Workshop lecture presented at the DiXiT Convention 2, Cologne, March 15. http://dixit.uni-koeln.de/wp-content/uploads/Vitali_Digital-formats.pdf.

Vitali-Rosati, Marcello. 2016. "What Is Editorialization?" *Sens Public*, January. http://sens-public.org/article1059.html.

Wals, Henk, Peter Boot, Herman Brinkman, Jan Burgers, Karina Van Dalen-Oskam, Sebastiaan Derks, Marc Dierikx, et al. 2012. "Onderzoeksprogramma 2012-2017." Den Haag: Huygens Institute for the History of the Netherlands.

Wark, McKenzie. 2001. "Essay: Codework." *American Book Review*, October 2001. http://amsterdam.nettime.org/Lists-Archives/nettime-l-0109/msg00197.html.

Weizenbaum, Joseph. 1966. "ELIZA – A Computer Program For the Study of Natural Languge Communication Between Man And Machine." *Communications of the ACM* 9 (1): 36–45. https://doi.org/10.1145/365153.365168.

Welch, Andrew. 2010. "Text, Image, Book: Reading William Blake — On Reading Blake's 'Milton' in Particular." *Imagining William Blake*. December 5, 2010. https://imaginingwilliamblake.wordpress.com/2010/12/05/blake-milton/.

Werkgroep Middelnederlandse Artesliteratuur. n.d. "Bartholomaeus Anglicus – Van Den Proprieteyten Der Dinghen (1485)." *Werkgroep Middelnederlandse Artesliteratuur*. Accessed July 5, 2018. http://wemal.nl/bartholomaeus-anglicus-van-den-proprieteyten-der-dinghen-1485/.

Westgeest, J.P. 2001. "De Leidse lapidariumfragmenten: delen van Maerlants *cortten lapydarys*?" *Queeste* 8 (1): 1–26.

Widdows, Dominic. 2004. "Word-Vectors and Search Engines." In *Geometry and Meaning*, 1 edition, 131–265. Center for the Study of Language and Information - Lecture Notes 172. Center for the Study of Language and Information. http://www.puttypeg.net/book/chapters/chapter5.html.

Wikipedia. 2020. "Google Docs." In *Wikipedia, The Free Encyclopedia*. Last modified 30 January 2020. https://en.wikipedia.org/wiki/Google_Docs#History.

Wimsatt, W.K. 1954. "The Intentional Fallacy." In *The Verbal Icon: Studies in the Meaning of Poetry*, and two essays written in collaboration with Monroe C. Beardsley, 3–20. Lexington: The University Press of Kentucky.

Winkler, Marieke. 2013. "Interpretatie en/of patroon? Over 'Het einde van de geesteswetenschappen 1.0' en het onderscheid tussen kritiek en wetenschap." *Vooys* 30 (1): 31–41. http://tijdschriftvooys.nl/index.php/2013/04/12/interpretatie-enof-patroon/.

Winter, Thomas N. 1999. "Roberto Busa, S.J., and the Invention of the Machine-Generated Concordance." *The Classical Bulletin* 75 (1): 3–20. https://digitalcommons.unl.edu/classicsfacpub/70?utm_source=digitalcommons.unl.edu/classicsfacpub/70&utm_medium=PDF&utm_campaign=PDFCoverPages.

Wittek, P., and W. Ravenek. 2011. "Supporting the Exploration of a Corpus of

17th-Century Scholarly Correspondences by Topic Modeling." In *Supporting Digital Humanities 2011: Answering the Unaskable*, edited by B. Maegaard. Copenhagen. http://www.clarin.nl/sites/default/files/sdh2011-wittek-ravenek.pdf.

Woolgar, Steven, and Geoff Cooper. 1999. "Do Artefacts Have Ambivalence? Moses' Bridges, Winner's Bridges and Other Urban Legends in S&TS." *Social Studies of Science* 29 (3): 433–449. https://www.jstor.org/stable/285412.

Wouters, Paul. 2004. "What Is the Matter with E-Science? – Thinking Aloud about Informatisation in Knowledge Creation." In *Society for Social Studies of Science Annual Meeting*. Paris. http://virtualknowledgestudio.nl/documents/_paulwouters/whats-the-matter-with-e-science.doc.

Wouters, Paul, Anne Beaulieu, Andrea Scharnhorst, and Sally Wyatt, eds. 2013. *Virtual Knowledge: Experimenting in the Humanities and the Social Sciences*. Cambridge, London: MIT.

Curriculum Vitae

English

Joris J. van Zundert (1972) was born in Goirle in The Netherlands. In 1990 he graduated from the St. Odulphuslyceum in Tilburg. After a period of working in IT retail and science communication, in 2001 he obtained a master degree in Dutch Linguistics and Culture from Utrecht University with a specialization in Middle Dutch literature. In the years following he worked at several institutes of the Royal Netherlands Academy of Arts and Sciences (KNAW). First as digital information expert at the Netherlands Institute for Scientific Information Services, and after that as a scientific programmer and researcher at the Constantijn Huygens Institute in The Hague and later at the Huygens Institute in Amsterdam. Currently he still works there as a scholar in the research group Computational Literary Studies and in the Digital Humanities Lab. He conducted his doctoral studies at the Leiden University Center for Science and Technology Studies and at the Huygens Institute in Amsterdam.

Nederlands

Joris J. van Zundert werd in 1972 geboren in Goirle. In 1990 behaalde hij het diploma vwo aan het St. Odulphuslyceum in Tilburg. Na een periode werkzaam te zijn geweest in IT retail en wetenschapscommunicatie, studeerde hij in 2001 af aan de Universiteit Utrecht in de richting Nederlandse Taal en Cultuur met een specialisatie in de Middelnederlandse Letterkunde. Hij werkte daarna achtereenvolgens bij verschillende instituten van de Koninklijke Nederlandse Akademie van Wetenschappen (KNAW). Eerst als digitale-informatiespecialist bij het Nederlands Instituut voor Wetenschappelijk Informatiediensten, daarna als wetenschappelijk programmeur en onderzoeker bij het Constantijn Huygensinstituut in Den Haag en later bij het Huygens Instituut in Amsterdam. Daar is hij nog steeds werkzaam als onderzoeker in de onderzoeksgroep Computationele Literatuurwetenschap en in het Digital Humanities Lab. Zijn promotieonderzoek deed hij bij het Centrum voor Wetenschaps- en Technologiestudies (CWTS) van de Universiteit Leiden en het Huygens Instituut.

ke
De dich
n. Het lijkt
ng met een besc
eligieus werk de on
benadrukt. De dichter m
de auteur van Madoc. Het hi
reclame, wat zou betekenen dat M
air werk was, waar hij succes mee
ng en Tekstkritische Aante
n de vroege Nederlands
guur van waarschij
We mogen aann
oon van e
n hem

9 789090 364605