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

Zundert, J.J. van

# 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 interdisci-

plinary 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 as-

pect 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 meta-tags that could be used for querying. Charles Goldfarb, a lawyer by training,

set out to develop Generalized Markup Language (GML). GML was a meta-language 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 biblio-

graphical 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 encod-

ing. 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 so-

ciety 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 no-

tions 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. How-

ever, 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.