



Universiteit  
Leiden

The Netherlands

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

Zundert, J.J. van

### Citation

Zundert, J. J. van. (2022, September 27). *Scholarship in interaction: case studies at the intersection of codework and textual scholarship*. Retrieved from <https://hdl.handle.net/1887/3464403>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/3464403>

**Note:** To cite this publication please use the final published version (if applicable).

## Chapter 6

# Author, Editor, Engineer: Code & the Rewriting of Authorship in Scholarly Editing<sup>1</sup>

“Programmers: stop calling yourselves engineers.”

(Bogost 2015)

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

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

---

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

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

## 6.1 What Is an Author?

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

Over the course of history the perception of authorship has been very tightly related to fluctuating ideas on who exactly produces textual meaning and interpretation. As a consequence the appropriation of authorship has been closely associated with an authority over epistemological claims, ontological claims, and with claims to truth. In contrast to classical, mediaeval, premodern, and even later times, today authorship seems to warrant far less authority over the interpretation of a text and the claims to truth it might make. In science and scholarship that authority arguably has shifted foremost to evidence and peer review. It has been eroded to a large extent by critic and

reader in the case of literature. This undermining of authority is however a relatively recent matter – and as we will see a contentious one.

As an historical overview much of the following borrows heavily from Burke (1995) who has described the history of authorship in the West as a continuous tension between the presence and absence of the author. Throughout this history views on authorship have swayed back and forth from author essentialism to almost complete impersonalization. Plato can be said to have sided with the idea of impersonalization. To Plato the personality and poetic imagination of an author was entirely unimportant. The world of forms was just a shadow of the ideal world of ideas, so any intervention by creative imagination on the part of the author when describing the world would just result in mere shadows of shadows. True knowledge could only be attained by disinterested rational enquiry. By advancing the idea of catharsis Aristotle on the other hand defends the empathetic nature of literature and poetry and thereby the empathetic role of the author. These perceptions of the role of the author are connected with epistemological claims. That is, essentially they are asking whether authorship is a means to knowledge – to which Plato answers “no” as it can merely record, and even that it can do only inadequately; but to which Aristotle answers “yes”, the empathetic imagination attached to the act of authoring may bear on knowledge. As such the roles of authorship and of the author become attached to epistemological claims, to claims about truth. With the rise of Christianity the debate about these roles becomes even more pivotal, as authorship pertains to the Scriptures and to hermeneutics, which is the theory of interpretation of these scriptures. How have the scriptures been conceived? Were the authors mere vessels to be filled with Divine inspiration, non-interfering bodies that merely moved the quill while sacred words passed through? And if this went for the scriptures, what of written interpretations such as those of Augustine? To explain the epistemological power of hermeneutics, the idea that divine truth can be revealed in an author is essential. This *auctoritas* becomes the keystone of the early mediaeval epistemology: an author that can be named and referred to as the authority for given knowledge. Yet, as the Middle Ages draw to a close a theory of hermeneutics emerges that holds that truth and knowledge may be acquired also through careful reading and reasoning. No longer are epistemology and truth given, instead they are now derived. Both epistemologi-

cal modes – that of given and derived – rely strongly on the presence of an author and the assertion of related biography. It is necessary to refer to an *auctoritas* to claim a truth with any authority. And in case a statement is not based on external authority, to be able to make claims on truth and truthfulness the moral and literary status of the author must be unquestionable.

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

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

The notion of possible subjectivity of interpretation only comes into the story with Albrecht Husserl’s phenomenology and Martin Heidegger’s ontological hermeneutics. These new theories of perception and interpretation erode the authority of the author. Late nineteenth century literary history and criticism were mostly concerned with establishing criteria of aesthetics – often motivated by a need for nationalistic literary canonization. Authorial poetics and the intent of the author were important aspects in this type of criticism. But the acknowledgement of the subjectivity of interpretation causes a “de-centering” of the author. This is reflected strongly in the New Criticism that accepts biographical information about an author as contextual evidence for a possible interpretation of a text, but generally assumes

that the true authorial intent cannot be established. In the words of Wimsatt and Beardsley in “The Intentional Fallacy”: “the author’s intentions in writing are neither recoverable nor pertinent to the judgment of the work” (Wimsatt and Beardsley 1954[1946]). From “The Intentional Fallacy” it is only a small conceptual step to Roland Barthes’ “The Death of the Author” (Barthes 1967). Here Barthes argues that interpreting a text solely along the lines of its author’s biography, is in fact limiting that text and the possible interpretations of it: “We know now that a text is not a line of words releasing a single ‘theological’ meaning (the ‘message’ of the Author-God) but a multi-dimensional space in which a variety of writings, none of them original, blend and clash. The text is a tissue of quotations drawn from the innumerable centres of culture.” In *S/Z* he would expand this idea into the notion of the “writerly text”, by which he means that any text is primarily a site for intertextual meaning production at the hand of the reader and critic (Barthes 1975).

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

Jeremy Hawthorn has argued that both Barthes and Foucault explored in these ways the possibility to read literary works with an attitude similar

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

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

Burke and Compagnon both call on cognitive and linguistic philosophy to argue that you cannot go around intent. That very faculty is applied to constrain the possible meanings of a text. No matter what subsequent context

dependent interpretations add to the polysemia of the text, in part the meaning that such an interpretation is derived from is shaped by the intent of the author. This is different of course than saying that there is a one to one relation between this intent, the text, and its interpretation. The author may have been unsuccessful in the expression, the reader may have been unsuccessful in decoding the expression, or indeed she may have “rewritten” the text reading it. The point of the matter is: intent is a necessary prerequisite for meaningful text.

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

Thus the report of the death of the author, her intent, and her authority was an exaggeration. On more precise inspection – e.g. the investigations of Burke, Compagnon, and Hawthorn – it seems that the death of the author was actually an epistemological metaphor rather than an ontological truth. The author needed to be decentered, even discarded, for a period of time for the post-structuralist debate to be able to develop the reader as a site of meaning production and interpretation (Burke 1989:57). However, having firmly established that subjective role it was time for the author to step back into the picture and claim back her own subjective role. This process is nothing more or less than an example of what Foucault argued in “What is an Author?” and to which that text itself contributed: that authorship is a socially constructed role (Foucault 1998:213). The role of the author and of authorship is brought in line with the purpose attributed to a text by its audience. Thus the lesser authority of authors on claims of truths in scientific texts and



the remaining to a certain extent of such in fiction. It is not either author or reader, but it is both of them negotiating the meaning and the interpretation of a text and the validity thereof from their individual situatedness. In the words of Compagnon: “The text [...] has an original meaning (what it means for a contemporary interpreter), but also later and anachronistic meanings (what it means for subsequent interpreters). It has an original signification (putting its original meaning in relation to contemporary values), but also subsequent significations (always putting its anachronistic meaning in relation to current values)” (Compagnon 2004:61).

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

## 6.2 The Author and the Editor

How does authorship relate to scholarly editing? Obviously both the creation of a literary or scientific text and the scholarly editing of the same – not uncommonly with several hundreds of years between those events – involve authorship. But what exactly is the authorship appropriated by the scholarly editor? Editing in general is the preparation of a text such that it may be printed or otherwise published for reception by an audience. Scholarly editing concerns itself with establishing, curating, and studying the record and archive of historical texts – in practice most often the legacy in some form of dead authors. If we follow Greetham (1994) on this the work of a scholarly editor entails at least the following activities: finding a text, reproducing the text (which involves reading, evaluating, and transcribing the text of the original document or documents), criticizing the text, and editing the text. This

process then results in a new representation of the text of the originating document(s). Although the phases of the process may be readily defined, each text is unique and each text confronts the textual scholar with complex and puzzling traces of its own genesis. In practice therefore the process of scholarly editing involves a series of choices that affect the eventual expression and representation of the text. Because of the many idiosyncrasies a particular text may hold, it is generally accepted that consistency and accountability should be the hallmark of a proper scholarly edition. Or in other words: the editor should be making the same choice in the same situation and explaining this to the reader – for instance, telling the reader that the editor chooses to represent the glyph “u” consistently as “v” where in a mediaeval manuscript that is used for the consonant.

Usually a scholarly editor will provide not just the conscientiously edited text, but considerable additional paratext as well. An introduction is often added arguing the text’s historical or cultural importance, or a specific peculiarity that motivates it being edited and republished in a scholarly fashion. The editor may put the text presented into context, explaining the genesis of the text, clarifying biographical details on the author, or expounding the text-theoretical underpinnings of the edition. Such additional knowledge may be part of the published text (e.g. the introduction) or it may be presented in additional scholarly articles, blogs, interviews and so forth. As an example we may point to Hans Walter Gabeler’s (et al.) edition of James Joyce’s *Ulysses* (Joyce 1984). Here all variants from the available manuscripts of the text have been painstakingly perused to aggregate an impressive synoptic edition reflecting the text as – according to the scholars – it was intended by the author. The same scholarly edition may serve to point out the predicament any scholarly editor puts himself in when creating a scholarly edition. Gabeler’s edition prompted a series of reviews and scholarly articles concerning its controversial editorial methodology (Lernout 2006). This debate has been regarded as a confrontation between Anglo-American and French-German editorial philosophies, but it also signifies a fundamental problem in philology and textual theory which is the assertion – as Dilthey would put it – that a text can at best be understood, but not explained (Compagnon 2004:41). The actual and exact intent of an author of fiction is in the end impossible to retrieve. Even what seems like a straightforward case of harmonizing orthog-

raphy – such as the distinction of u/v – results in the undecidable pondering of ambiguous expressions (cf. for instance Gerbenzon 1961). The reader who is not a medievalist might not know, but the glyphs ‘u’ and ‘v’ have been used interchangeably at times and sometimes it cannot even be decided if one or the other was meant by a copyist of a medieval or classical manuscript, let alone that it would be decisively decidable what the original author intended to write. If even such “trivial” cases are hard, what about the interpretation of the work as a whole? This is where the practices and principles applied in many scholarly editions clash with ideas on the subjective interpretability of text and the impossibility of establishing authorial intent (Wimsatt and Beardsley 1954[1946]). Ultimately we cannot know with any certainty from the text or the biography the intent that is part of an author’s transcendental ego. As both Burke and Compagnon argue we run the risk of making anachronistic interpretations of a text because of our distance to the time and context of the author. And we may be fooled because the text is an inadequate expression of the governing intent of the author. All editorial decisions are thus at risk of not (well) representing what was intended by the author. All editing is interpretation, an estimation of what the text was intended to express. Scholarly editing then, lays a claim to a truth, which is that the delivered edition is the most warranted expression of the text by scholarly standards. Scholarly editing is not the same as the authorship it claims to represent, it is however authorship that claims to represent that authorship. In this respect the authorship involved with creating a scholarly edition is very close to Harold Love’s definition of revisionary authorship (Love 2002:46), so that in effect the editor becomes the self-asserted proxy of the author. A cynical interpretation would cast the scholarly editor as an imposter exploiting the text of an author. Indeed the image of the “editor [...] as a brutish interloper forcing his interpretation on the defenceless text” led Murphy (2008) to muse: “Much of the most advanced contemporary theorizing about editing has suggested that the editor should follow the author into his grave”.

Broadly speaking scholarly editing formulated two responses to this post-structuralist challenge. One response was to fully embrace the ideas of the reader as the site of production of interpretation. In Greetham 1994, David Greetham’s rejoinder to the post-structuralist challenge was

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

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

The other response was, rather in contrast, a reinforcement of the primacy of

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

Well-known practitioners and theorists of textual scholarship such as Hans Walter Gabler and Elena Pierazzo likewise retreat forcefully to the primacy of the document. Robinson commented: “For Pierazzo, the possibilities of the digital medium have created new possibilities, which enable the making of detailed digital representations of the document using complex encoding. [...] For both Gabler and Pierazzo, digital editing is rooted in the document. [...] It is difficult to imagine a more articulate and forceful exposition of a theory of digital editing as focused on documents than that given by Gabler.” (Robinson 2013a:110–111). Together with McGann these scholars assert that the materiality of the document allows a claim to a truth as philologic fact – and with this simple gesture McGann reduces the impact of hermeneu-

tics and the pertinence of authorial intent for the text to mere afterthought: “For the philologist, materials are preserved because their simple existence testifies that they once had value, though what that was we can never know completely or even, perhaps, at all. If our current interests supply them with certain kinds of value, these are but Derridean supplements added for ourselves” (McGann 2013:346).

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

must impose their truths before others impose their fictions” (Parry 1971:413 [1936]).

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

Thus both responses by textual scholarship to the post-structuralist challenge attempt to evade as much as possible a claim to the interpretation of the text – for the more interpretative an edition is, the more the scholar asserts to have knowledge about authorial intent. Yet, a retreat to philological fact does not resolve the predicament. If the genetics of a text are not difficult at all, the role of the scholarly editor is trivial. But if they are hard – such as in the case of the Joyce edition of Gabler, or in the case of intricate phylogenetics of mediaeval manuscript traditions (cf. e.g. Reenen et al. 2004) – much interpretation is required to establish the text. But strong interpretation comes at the risk of anachronistic and subjective readings. Nor does the “archive”, if understood as an (electronic) archive of “philological facts”, resolve the issue. For the same holds – in extremis it would be a repository of (digital) image

facsimiles, and again the role of the editor would be trivial. Moreover, it is questionable how fair it is to leave such an archive to the reader without any guidance. The skills, experience, and expertise of the scholarly editor would be withheld from the reader. For this reason Murphy argues for strong interpretation: that editors must presume to know what readers ought to know (Murphy 2008:306).

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

Supposedly objectified material and philological fact-based scholarly editing has developed into a dogma. To elaborate on Rorty: no doctrine is dangerous, but the thought that textual scholarship depends solely on one doctrine – that the hand of the editor can behave such that it does not taint the meaning of the text – is. Any editing is an ‘intrusion’ on authorial intent. Even the question whether a deceased author would have wanted the text to remain in existence cannot be satisfactorily answered by philology. The more sensible role for philology is thus interpretation – interpretation that cannot in any case be circumvented. It must therefore always be clear that the text



presented is an interpretation. We should not be made to think that we are reading James Joyce's *Ulysses*. The reader ought to know that she is reading *Ulysses according to Gabler*.

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

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

new scholarly edition, which paradoxically introduces all the same problems of authorial intent in a recursive fashion.

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

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

Foucault argued that the authority of the "biographical" writer (i.e. the author and his intent) on interpretation was less in the case of scientific texts. Interpretation in this genre was mostly the prerogative of the discourse and those involved with that as a whole. The author's name was but a handle to the text (Foucault 1983:12–14). For Foucault the appropriation of the text

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

### 6.3 The Author and the Engineer

Scholarly editing is the appropriation of revisionary authorship. Its aim is to make a scholarly motivated contribution to the textual “archive” – which

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

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

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

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

//End -->
</Script>
```

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

.. my.time: my time: it\_c(wh)or(e)por(ous+h)ate\_ \_experience\_\_\_\_he(u)rtz\_\_\_\_.] [end]

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

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

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

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

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

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

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

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

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

Code creates a multi-dimensional space of structure and meaning, just as text does (Huitfeldt 1995). Following Compagnon as in the previous section, a text may have contemporary meanings and anachronistic ones; it will keep on creating new meanings over time, as long as it is read. Programmers re-use code. They re-use their own and they re-use code that they find in “code libraries” (programming idiom is sometimes pretty self-explanatory) of other programmers. Barthes’ observation, that a text is not a line of words releasing a single meaning but a multi-dimensional space which recombines a variety of writings (Barthes 1967), is as true for code as it is for text. In a similar vein Friedrich Kittler calls on Derrida (Kittler 1993a:225–226) to argue that the creation of the digital form of text that code is, is indeed nothing more or less than the continuation of writing with different means. But he also points

to the essential difference between the two sorts of writing: that digital writing is “im Unterschied zu allen Schreibwerkzeugen der Geschichte auch imstande [...] selber zu lesen und zu schreiben” (Kittler 1993a:226). Differently put, the difference between text and code is that of the difference between performance and auto-performance. A print text cannot itself perform, it needs a reader to accomplish that. Digital text, code, can perform. It can even self-perform, though this is seldom the technical reality; in the vast majority of cases it is read and executed by another piece of software that follows its instructions, much indeed like a Jacquard loom (Ceruzzi 2012). Donald Knuth realized that these different natures of code as on the one hand machine instructions and the other hand meaningful text mixed two types of literacies: the literacy of the executable code and the literacy of non-executable text. This led him to develop a computer system that explicitly caters to these different literacies, although his direct motivation was pragmatic: “I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: ‘Literate Programming’” (Knuth 1984:97).

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

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



symmetrical. Code can operate on text, which results in the same, new, or transformed text:

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

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

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

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

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

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

---

<sup>2</sup>[https://sourcemaking.com/design\\_patterns](https://sourcemaking.com/design_patterns)

<sup>3</sup><http://shop.oreilly.com/category/series/cookbooks.do>

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

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

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

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

---

<sup>4</sup>Cf. <http://www.textgraphs.org/>

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

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

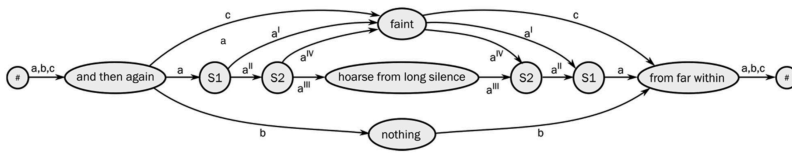


Figure 6.3: Graph model of genetic stages in a text (Haentjens Dekker et al. 2015).

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

Also software and its creators now become part of this interpretation. Bauer (2011) in answering a recurring question on the perceived lack of theory in coding and digital humanities practice quotes Susan Smulyan calling out: “The database is the theory! This is real theoretical work!” Galey and Ruecker (2010) have argued similarly that design, prototypes, and code can be scholarly arguments as well. Further examples – such as the

understanding of coding as a form of disciplined play and in that sense as an instrument of research (Rockwell 2003), and reflections on the hermeneutic role of code in the digital humanities (Ramsay 2011b) – support the view of coding as a form of scholarly authorship and argument as well.

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

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

---

<sup>5</sup><https://github.com/>

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

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

## 6.4 Being Critical About Code

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

---

<sup>6</sup><http://vangoghletters.org/vg/credits3.html>

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

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



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

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

## 6.5 Conclusion

Scholarly editing and the creation of code that is part of that process are both forms of revisionary authorship that contribute through scholarly argument to a hypothetical expression of a text. It would therefore be academically irresponsible not to develop a methodology that can acknowledge the scholarly contribution and status of the creation of digital objects and code, and

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

---

<sup>7</sup><http://www.elsewhere.org/journal/pomo/>

<sup>8</sup><http://runme.org/feature/read/+postmodgen/+57>

