**VAN ZUNDERT — When?**

<0 figures>

**When Is Coding Scholarship and When Is It Not?**

Van Zundert, J. J. and Haentjens-Dekker, R.

We argue that the humanities and digital humanities need to consider certain forms of code as scholarly object and certain types of code authorship as a scholarly activity. It will be essential, therefore, to develop a scholarly mode of evaluating and criticizing these scholarly contributions in code. We argue this position from a theoretical perspective, and we present a case study to consider the boundaries of scholarship in code and coding, and how this scholarship in code may be evaluated.

Software, code, and algorithms are becoming an inherent part of the humanities, and not just because they are tools within digital humanities. Rather, code and software have resulted in the softwarization of society as a whole (Berry, 2014). The process of the digital becoming ubiquitous—of code and algorithms permeating culture and society in a radical manner—is not a new phenomenon (cf., e.g., Coyne, 1995; Capurro, 2010). However, in the last decade the emergence of digital technology in every aspect of culture and society, and in every associated workflow, seems to have accelerated vastly with the advent of mobile computing and social media—so much so that Steve E. Jones (2014) in his recent history of the digital humanities compares this development with the ‘eversion’ envisioned in the 2007 novel *Spook Country* by William Gibson: a turning inside out of digital technology intertwining with the physical world and real life to the point of complete and indistinguishable mergence. Inter alia, David M. Berry (2014) concludes that society as a whole has therefore moved into a post-digital era. For the unsuspecting or impartial user, this means that a perfectly transparent layer of omnipresent digital technology sits between digital resources, data, and information and the uses or experiences generated from these data and resources. The invisibility or—if one prefers—covertness of this transparent layer creates an illusion of neutrality of this very layer, arguably reinforced by code and software being grounded in logic and mathematics that lend digital technology an aura of impartiality and not seldom even infallibility. However, code and software are non-neutral and fallible (cf., e.g., Coyne [1995] and Berry [2014]). Being manipulable, being engineered, and being authored by people, code has meaning, intent, assertions of truth, and performative aspects of and by itself.

Given the radical ubiquitousness of a digitality that is seemingly transparent but potentially non-neutral, transformative in virtually every aspect of culture and society as well as in scholarship, one would expect the humanities on principle to be heavily involved in research towards the properties and effects of the innards of this digitality. In other words, one would expect the humanities to be interested in the scholarly methodological merits and effects of code and code engineering. Until now, however, code and code engineering have hardly been researched as to their scholarly significance and impact. Code as scholarly object and code authoring as scholarship have been less a point of attention of the humanities than the application of resulting software. This focusing away from code is also rather mainstream within the digital humanities. The debate on code literacy—does one need to be able to code to call oneself a digital humanist?—is undecided but leaning mostly towards asserting that code skills are not a necessity for digital humanists (Ramsay, 2011). The propensity to foreground the results of applying digital or computational analysis and to discuss especially those results on a scholarly level, rather than making algorithms or code the object of scholarly discussion, is also apparent in the writings of practitioners of code in the humanities: mostly what has been done with software and what the results might mean are highlighted, far more than the problematizing of the very digitality of method (cf., e.g., Ramsay and Rockwell, 2012). Because of the argued increase in prominence of digital technology, however, neither the humanities nor the digital humanities can forgo finding a genuine scholarly mode to disclose code and coding objects, and to develop scholarly critical mechanisms for such objects.

Thus, because of the ubiquitous role of digital technology both in society and scholarship, we argue that humanities cannot evade taking code and code authorship into its domain as objects of study and criticism. Often in answer to this problematic issue, one hears a hand-waving argument made that one does not need to know how a combustion engine works to drive a car. We argue that this is a false and, worse, dangerous metaphor. Ian Hacking (1981) uses the same metaphor when arguing that one needs no deep theoretical understanding of optics and the physical laws that predict the refraction of light or the behaving of electrons when operating an optic or electron microscope. In the case of microscopy this is valid, as these laws of physics have been widely theorized and heavily empirically supported. Given our scientific consensus expectation that our theory of optics will thus hold, the technology of a microscope can be blackboxed in a true Latourian sense (Latour, 1988). Software code, however, is not governed by laws of physics. Although code has to answer to certain logic to be executable, it need itself not even be logical or mathematically precise in effect, reasoning, or operation. Certainly, as long as code applications in the (digital) humanities remain a ‘parade of prototypes’ (Wouters and Beaulieu, 2007), code engineering will hardly be ruled by generic laws. Even if the mathematics of underlying analytic software may be profoundly formally tested, one is hard-pressed to find software applications in the humanities that are accompanied with formal test suites guaranteeing the correct integration of these mathematical components. Especially the heuristics of purpose-built software should be subject to rigorous automated testing, and to scholarly scrutiny and review. We argue that to improve the situation it is essential to develop a techno-scholarly mode of evaluating and criticizing scholarly contributions in code. Some work has been done in this area, most notably by Montfort et al. (2012) and, e.g., Mark Marino (2006). The literature so far, however, considers code as object external to the humanities. But how do we respond to Jean Bauer’s (2011) echoing of the call that ‘the database is the theory!’? Confirming code as a scholarly object, evaluating its functions, and acknowledging its production as scholarly activity require a theory and critical framework that have yet to be formulated.

Such a framework should be built on the basis of practical approaches to the criticism and the evaluation of code as scholarly object. The remainder of our paper will therefore underpin our theoretical exposé with a concrete case study of the development and the code base of *CollateX*. *CollateX* is a successful text collation engine developed in a digital humanities context (Haentjens Dekker et al., 2014). The development, algorithm, and logic of *CollateX* marries code engineering skills with mathematical methods, computational algorithm design, and decidedly scholarly heuristics. We demonstrate how certain scholarly decisions of collation can be expressed as mathematical problems and logically programmable code, but that in other cases it is essential to hand off decisions to scholarly interpretation. Determining the precise cutoff point between the application of mathematics or logic and the application of scholarly expertise is, as we demonstrate, a skill that is informed by both high-level computational as well as high-level scholarly expertise. We thus also demonstrate from practice that code has intrinsic scholarly purport and intent. Lastly we consider how this purport can be systematically scrutinized by applying both automated testing and a scholarly critical approach.

We then return to the overarching concern of evaluating the scholarship in code, and we argue that ignoring the scholarly nature of code renders humanities willfully illiterate of a technology that is re-inscribing scholarly content and practice.

**References**

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

**Berry, D.** (2014). *Critical Theory and the Digital*. Critical Theory and Contemporary Society. Bloomsbury Academic, New York.

**Borgman, C. L.** (2007). *Scholarship in the Digital Age: Information, Infrastructure, and the Internet.* MIT Press, Cambridge, MA.

**Capurro, R.** (2010). Digital Hermeneutics: An Outline. *AI & Society,* **35**(1): 35–42.

**Coyne, R.** (1995). *Designing Information Technology in the Postmodern Age: From Method to Metaphor.* Leonardo Book Series. MIT Press, Cambridge, MA.

**Drucker, J.** (2003). Designing Ivanhoe. *Text Technology,* **12**(2): 19–41.

**Hacking, I.** (1981). Do We See through a Microscope? *Pacific Philosophical Quarterly,* **62**(4): 305–22.

**Haentjens Dekker, R., Van Hulle, D., Middell, G., Neyt, V. and Van Zundert, J.** (2014). Computer Supported Collation of Modern Manuscripts: CollateX and the Beckett Digital Manuscript Project. *Literary and Linguistic Computing,* March, doi:10.1093/llc/fqu007.

**Jones, S. E.** (2014). *The Emergence of the Digital Humanities.* Routledge.

**Latour, B.** (1988).*Science in Action: How to Follow Scientists and Engineers through Society.* Harvard University Press, Cambridge, MA.

**Marino, M. C.** (2006).Critical Code Studies. *Electronic Book Review*, http://www.electronicbookreview.com/thread/electropoetics/codology.

**Montfort, N., et al.** (2012).*10 PRINT CHR$(205.5+RND(1)); : GOTO 10*. MIT Press, Cambridge, MA, http://10print.org/.

**Ramsay, S.** (2011). On Building. 11 January, http://stephenramsay.us/text/2011/01/11/on-building/.

**Ramsay, S. and Rockwell, G.** (2012). Developing Things: Notes toward an Epistemology of Building in the Digital Humanities. In Gold, M. K. (ed.), *Debates in the Digital Humanities*. Minneapolis: University of Minnesota Press, pp. 75–84, http://dhdebates.gc.cuny.edu/debates/text/11.

**Wouters, P. and Beaulieu, A.** (2007). Critical Accountability: Dilemmas for Interventionist Studies of e-Science. *Journal of Computer-Mediated Communication,* special issue, **12**(2): 583–99.