

SOFTWARE STUDIES METHODS

Matthew Fuller

What is the consistency of computational culture? What are the ways in which the objects of computer science and its more informal offspring operate in the world? Looking into such issues, it can be readily seen that a cunningly diffuse and multilayered aspect of the question of method is understanding what constitutes the problem. Luckily, for the development of a field such as software studies, computational media cultures are abundantly generative of such things. Problems flourish. From questions about the foundations of computing to the articulation of cultural tendencies and political systems across commodity platforms, via the questions of power, aesthetics, and processes of subjectivity, problems abound, and that is only to mention a few that are properly named as such. Any short survey of methods in the field can therefore only be partial and indicative, picking out a few general trends. The kinds of problems addressed arise, too, at the emergence of software studies at the beginning of this century. However, from such a starting point, software studies has become the grounds for a diversity of methods, concerns, and conceptual and practical resources. This chapter maps a few of these and give some sense of their consistency and trajectories. In this, as a survey, it will necessarily be limited due to the vitality and range of activity in the field. Read it, then, as an aperitif, before investigating further.

To frame the question of the problem, a little genealogy is in order. From a certain set of angles, software studies arises from a background of bemused frustration at the ways in which “high level” media theory would tend toward subsumptive generalization about the “virtual,” or about “cyberspace,” without attending to the howl and screech of a dial-up modem or the particular dance of pixelated entities among the Graphical User Interface (GUI) that constituted everyday activity in the “knowledge economy.” A parallel here can be drawn with software art, which articulated itself in part against the media art and net art that often ignored the specific materials from which it was made, a condition that—as those materials became both increasingly interesting and increasingly overcoded—became untenable. Equally, as social theory drew upon the active constitution of society by systems, media, and tools, alongside those entities that had hitherto been understood to be social, the capacity to recognize what Scott Lash (2002) called “Technological Forms of Life” became pressing. Amid this, too, the imperatives underlying the development of the field are driven by an emphasis on the complexly materialist, fortified by the empiricism of the abstract found in both the more generous reaches of poststructuralist legacies and those aspects of meta-mathematics and computer science that had always recognized the cultural dimensions of their activity.

More recently, software studies finds itself in the curious position of being a scholarly stick-in-the-mud as certain aspects of computational culture become sprinkled like glittery mica on trays of theoretical cupcakes: decorative, indigestible, and garnishing the semiotically retrospective and materially stale. For instance, algorithms are increasingly emphasized as an explanatory actor, although in a surprisingly large quantity of scholarship this explanatory power is asserted without their contextualization within other systems in which they are embedded, and even without ascertaining what an algorithm might be. A scan of the literature shows that it is entirely possible to have a social or literary theorist discourse on “algorithms” without any references other than to others of their ilk. This kind of elective solipsism of disciplines is familiar to those navigating interdisciplinary terrains and is also articulated through the kinds of citation politics that other such fields, such as feminist research, have become all too familiar with (Fábos & Haddad 2013). We are in the amusing position where the emphasis on materiality is elevating technical content to the same kind of generalizations that, say, the more banal pronouncements of postmodernity suffered from in their heyday. Social theory, too, often brings along its upstairs–downstairs split between high theory and empiricism.

There is an art to the interplay between the rigors imposed by attention to material qualities and conditions with all their constraints, limits, and capacities; their specificities and individuations; and those entailed also by the particular epistemic and ontogenetic capacities of abstractions. To address such topics, software studies approaches might characteristically tend to identify specific algorithms, articulate their genealogy, recognize and work with their characteristics, and see them as part of a larger assemblage—one that is not in turn immune to the adventures of conceptual rigor (see Mackenzie 2010). All this requires something of an avidity for research. Such a condition also drives it toward noncanonical texts and sites as sources of interest. This drive in turn is a methodological choice that sets it at a tangent toward theoretical work aiming to array itself solely among well-cited or paternalistically approved books. Showing that one is in dialogue only with those of the correct lineage is a rhetorical technique that has the democratic advantage that the canon is seemingly available to us all, whether we like it or not; but it also may be affected as a technique of foreclosure. Part of this condition is due to the febrile pleasures of academic politics, in which it is assumed that the study of new media has yet to fully establish itself as a legitimate field and, to do so, must exaggeratedly exhibit the habits of conformism and deference special to the various overlapping pyramid schemes that are, among other things, operative in the university.

Another way to frame the question of methods is to partially arrive at it via two fields that contribute to software studies. In different ways, cultural studies and computing both emerged in the aftermath of the Second World War, forged by the necessity for an interdisciplinary bringing together of multiple methods, materials, and problematics. In different times, software studies emerges as a minor mutational fold within these discourses, at the cusp of the inventive enthusiasm of the world wide web and the internet, and the transition to the version of the same which we now inhabit. The various crises and possibilities in knowledge, technology, institutions, and epistemic potential that each of these represents feed into the specific traits of software studies as a field.

With cultural studies, software studies shares a taste for vulgar objects that pass below the threshold of critical perception due to their mundane or tedious nature. Ostensibly boring entities, manuals, slide presentations, and technical specifications are all probed for interesting traces as admixtures of high technology and low theory. Just as crystals grow differently in variable electromagnetic fields, which can in turn be examined through the variations in

growth patterns to which they contribute, the relatively “pure” formalisms that software articulates in such materials can also retain interesting patterns of resonance with wider formations.

Relatedly, one may sense into software by the ways it comes into composition with users. And here, ignorance of software’s conventions has been highly revealing. It must be possible to read software in a way that is partially naïve, stupid, heading in the wrong direction, mysterious, uninformed by a disciplinary history. To watch a person encounter a machine for the first time is exciting; one might learn from their dissonant interpretation what experienced preconceptions have become. Unfortunately, even with the confusions of users moving between operating systems, today the state of inexperience with computational entities is a rare resource. But there are ways in which it is mobilized, bearing indeed some relation to Jacques Rancière’s formulation of the “ignorant,” where people learn by moving along in a manner in which “one shouldn’t move along—the way children move, blindly, figuring out riddles” (1991: 10). Indeed, given its value, ethnographical techniques have been utilized to capture this moment, or to set up a pretext in which it can be removed from the stories of products (Wilkie 2010). There is a kind of sense in which this naivety, as a prerequisite for the genesis of thought, may seek new objects of encounter.

One way such naivety may fruitfully manifest is by asking scrupulously simple questions. Indeed, the cunning and intelligence required for the deployment of a simple question is a key concern for programmers attempting to instruct machines, and for anyone working to elicit material from what is now characterized as big data (Kitchin 2014a; Elmer, Langlois & Redden 2015). The artful projects of the working groups convened by Lev Manovich, such as *Selfie City* (Software Studies Initiative 2014), work in this mode, and one can say that they operate at a number of levels. Ostensibly, they are navigations of certain sets of data. A question is asked of a flow of images that conform to a certain categorical norm, that of date, kind, genre, keyword, location, and so on. These categorical terms are those that arise out of the specific nature of the system, which is partly that of a database whose genealogical roots are in set theory. Corralled as it often is within categorical architectures and tucked behind Application Programming Interfaces (APIs), it is difficult to get significant traction on the material, except in reconfirming its preconditions.

To work around this, data are captured and *withdrawn* from such systems and then subject to other kinds of query: for instance, using color saturation to navigate variations in images tagged with the names of certain cities, or image-recognition software is used to analyze the tilt of the head characteristic of selfies in a certain location. The findings of such work are partly in this activation of the parodically positivist formalisms that undergird computational culture and subtend big data. Genuinely trivial details in aggregate become something else; they constitute part of cultural form. To navigate these and aggregate them in turn becomes part of a dual capacity of critical exploration and seamless involvement that is both troubled and fascinating.

Working through the ways in which computational forms change established modes of culture, politics, and society and establish new ones has—like the traces of forces acting on a crystal—been very palpable in certain cultural-economic forms, such as publishing, music, and film, where they have had massive systemic effects. In other disciplines, such as philosophy, the articulation of these predispositions and capacities has been less immediately discernable but also intriguing, with the development of platforms and the mobilization of previously more marginal actants, but also cognitive speed-ups and conceptual hazing becoming more visible. Here, computational forms become one of the fields in which the crystal grows.

Related in certain ways to the question of operative stupidity and the principle of the pyramid scheme in academic citation patterns is the entrenchment of Pareto-principle-type citation patterns due to the material specificity of contemporary publication (Becker & Stalder 2009). When the search engine becomes the primary basis of scholarly research, the keyword becomes a crucial pivot for the articulation of fields. Thus, in speeded-up research cultures, ever-ready to pronounce what passes for a novelty, it is possible for communities working in related domains to entirely bypass each other's work due to over-dutiful following of links.

Earlier I noted that software studies methods include something of the fastidious, pains-taking work of the archive following a specific object. Here, there is a useful correlation with work suggested by Kopytoff (1986) and in the figure of the boundary object formulated by Susan Leigh Star and James Griesemer (1989). A key factor to note is that many objects in software studies are multiscalar. To fix only on the immediately empirical—say, on the position of a specific procedure within a wider assemblage—would also be to risk missing, via simply descriptive means, some of the more abstract dimensions of such an assemblage. The mathematical and logical conditions of such an entity are also inherently formed among non-uniform fields of metaphysical “radiation” set in motion by mathematical conditions (Fazi & Fuller 2016). Equally, their articulation can be probed in relation to the epistemic conditions of certain kinds of data cultures and economies. To recognize the multidimensionality of the problem is by no means to deny the pleasures or necessity of micro-scalar analyses, but rather to emphasize the ways in which positing an analysis of a part triggers in turn the positing of a putative whole.

To return, then, to the relation of the method to the problem, part of the work of software studies has been to burrow into and articulate the relations between different scales in which software becomes manifest without proposing any pre-existing, hierarchically ordered set of conditions to which such things must correspond, without at the same time failing to notice that such orderings are multiple and highly operative. To this end, there is a discussion of software in terms of the data-structures known as “stacks” in the work of Ben Bratton (2016) and Rob Kitchin (2014b), among others, examining the ways in which the mutual ordering and dependencies of separate scales of abstraction and operation produce functional wholes. When, via an overly normalized and normative empiricism, work in this domain risks a simple recapitulation of systemic technical description, it is also challenged by a materialism operating “below the stack” by, for instance, focusing on the social and even alchemical histories of the minerals that end up in dialogue with software in the work of artists such as Jonathan Kemp, Martin Howse, and the group YoHa. Both of these tendencies afford further dialogue with wider questions of infrastructure and the scholars, from Sergio Bologna (1991) to Keller Easterling (2014), who have traced these inter-relations via political economy and spatial approaches.

Questions of infrastructure appear, too, in much of the work on understanding the fundamental components of large-scale web platforms. There is a sense that a key movement in much contemporary work is an investigation of the operations, significance, and underlying forces and consequences of social media and the apparatuses of production, storage, dissemination, and analysis that they constitute. Numerous articles have seen scholars address the functioning of Google's PageRank or Adwords; Facebook's OpenGraph; the shifting nature of database systems, from relational to NoSQL; and other mechanisms that deftly bring together theoretical resources drawn from different strands of cultural and social theory as well as histories of organization and mathematics (Dourish 2014; Kaldreck & Röhle 2014).

Treating the entities of such “abstract infrastructure” as inherently cultural, social, and political and also interrogating how computational media install and format modes of being and becoming in the present are fundamental to the genesis of software studies methods.

One of the conditions here is the changing status of media. Friedrich Kittler neatly boiled down the condition of media as being those mechanisms to do with the production, storage, and dissemination of information. The systems in the last paragraph added *analysis* to this triad. Here, analysis is the breaking down of complex entities into what, at a certain scale, can be read as nominally fundamental units, and working out their immanent, potential, or emergent relationships. Media have in many cases become a subset of computational systems. As a condition of their constitution as digital systems and, thanks to Alan Turing, as procedural systems that are inherently composed of discrete entities and steps, computers are constitutionally predicated upon analysis. Analysis and control combined are part of what make the move to computational generality such a significant, if not unprecedented, shift. It is in how this power of computational analyses is coupled with other forms native to the arts and humanities that we should now turn.

The collectively authored book, *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10* (Montfort et al. 2014), proposes that the single line of code set out in its title can be read, rewritten, divagated from, and used as the basis for memory and reverie in a compelling way. Here, close reading is the pivot around which a world can be spun out from a crystal of code. In related terms, a number of similarly minuscule objects have been examined for how their minor variations can become highly revealing entry points into a wider set of phenomena. Anne Helmond’s (2013) discussion of the historical changes to the hyperlink or URL—from something hard-coded into an HTML document to an artifact generated on the fly by a database system—exemplifies this approach. By looking at such changes, insights into how websites are used and embody different modes of social and cultural activity can be readily elicited—for instance, in the trajectory from something hand-coded and available to all users to being an ostensibly secure, one-time only transaction. Related articulations of such telling detail are evident in Ben Grosser’s (2014) account of his plug-in, the “Facebook Demetricator,” which removes numerical and temporal data from the interface of the social media site. Another recent approach is to combine analytical methods from police data forensics with a survey of software manuals to articulate a history of metadata for author information (Fuller, Mazurov & McQuillan, forthcoming). In these and many other projects, there are numerous contributions to debates about what constitutes an object of study, entangled with the methodical means of staging such an encounter. Work by Wendy Chun (2011), Alex Galloway (2012), Annette Vee (2012), Taina Bucher (2013), Shintaro Miyazaki (2012), and numerous others propose methods integrated with particular lines of inquiry around certain scales at which software exists.

An additional mode of analysis, drawn from technical practices, is that of reverse engineering. Here, Adrian Mackenzie suggests:

One strategy is to begin by describing the most distinctive algorithmic processes present, and then ask to what constraints or problems these processes respond. From there we can start to explore how software transforms relations.

(Mackenzie 2008: 50)

Such a methodical frame corresponds with the process of observation and decomposition adopted in Robert W. Gehl’s (2014) research on social media. It also provides an impetus

for working directly on software systems via programmatic comparisons between input and output (Feuz, Fuller & Stalder 2011; Skeggs & Yuill 2015). Such research involves setting up one system to interrogate another. Applied to a target such as predominant web-based services, this interrogation may involve establishing user accounts, scripts, and servers that feed data into a system to generate results and then make comparative analyses of them. Sometimes described as an “algorithm audit” (Sandvig et al. 2014) and strongly developed by researchers such as Latanya Sweeney (2013), these techniques are especially notable in formulating a progressive program of research on the empirical operations of computational systems that constitute privatized forms of public resources.

One of the difficulties here is negotiating the privileged position that quantification is allocated in different kinds of discourse, where the numerically describable stands in for the empirical. Such quantitative research is often that most embraced and, if not actually discussed, replicated in the press. Nevertheless, as a means of reflexively working through how computational forms operate, it is an intensely valuable set of approaches. In a context where massive amounts of economic, social, and cultural actions are processed in computational environments, quantitative research is a mode of action conducted on a tactical basis by multitudes of actors, including algorithms employed by competing companies to understand the operations of others. We can say, too, that as reverse engineering becomes a mode of sociability—for instance, one prods a person to see if they elicit the signs taken for friendship—it is also a mode that becomes increasingly omnipresent.

Methods result in many cases from the conjunction of an intensive question with a problem. A case that is interesting to the point of fascination demands subtle attention to cracking it. Here, the mode of study adequate to a digital entity or process achieves the tender revelations of a hack. Exemplary in this regard are the edited volume and exhibitions of artworks put together by Olga Goriunova (2010, 2011, 2014) on the theme of *Fun and Software*. One of the guiding concerns in this strand of Goriunova’s work is to draw out the ways in which software is attached to obsessive pleasures of multiple kinds. This series of projects draws rich attention to the articulation of logics, processes, systems, and other modes of computational artifact as they are not only manifest in—and given grounds and shaped by—the nature of software but also provide a condition of life that is both sensual and abstract. Crucial to note in the *Fun and Software* series is that scholarly modes of research are challenged by those that are anterior to the question of academic form. The condition of software is understood to be fundamentally existential. Aesthetics provides routes into such a condition, but so also do the reflexive accounts of cultural and technical practitioners. There is a close involvement here, too, with the immanent mode of criticism established in software art and the properly speculative dimension required when thinking about the various scales of abstraction and concretion found in software cultures. (Related approaches such as design fictions and speculative design are important to map here.) Here, software studies develops close affinities to the philosophical approaches of writers such as Luciana Parisi (2013) that mobilize reflections on the ontological conditions of mathematical technologies.

Following from this, the question of method is one coeval with multiple modes of being as much as with the problems into which those modes are crystallized. Appropriate methods involve grinding up such crystals to see what they do when ingested, describing what they mix with and the forms they take, or gazing deeply into them to divine the nature of the present.

Further Reading

- Computational Culture: A Journal of Software Studies*, retrieved from www.computationalculture.net.
 Ekman, E. (ed.) (2013) *Throughout: Art and Culture Emerging with Ubiquitous Computing*, Cambridge, MA: MIT Press.
 Manovich, L. (2013) *Software Takes Command*, London: Bloomsbury.
 Ulrik Andersen, C. and S. Bro Pold (eds.) (2011) *Interface Criticism: Aesthetics Beyond the Buttons*, Aarhus, Denmark: University of Aarhus Press.

References

- Becker K. and F. Stalder (eds.) (2009) *Deep Search: The Politics of Search beyond Google*, Vienna: StudienVerlag.
 Bologna, S. (1991) "The Factory-Society Relationship as an Historical Category," trans. E. Emery, retrieved from libcom.org.
 Bratton, B. (2016) *The Stack: On Software and Sovereignty*, Cambridge, MA: MIT Press.
 Bucher, T. (2013) "Objects of Intense Feeling: The Case of the Twitter API," *Computational Culture* 3, retrieved from computationalculture.net/article/objects-of-intense-feeling-the-case-of-the-twitter-api.
 Chun, W. H. K. (2011) *Programmed Visions: Software and Memory*, Cambridge, MA: MIT Press.
 Dourish, P. (2014) "NoSQL: The Shifting Materialities of Database Technology," *Computational Culture* 4, retrieved from computationalculture.net/article/no-sql-the-shifting-materialities-of-database-technology.
 Easterling, K. (2014) *Extrastatecraft: The Power of Infrastructure Space*, London, UK: Verso.
 Elmer, G., G. Langlois and J. Redden (eds.) (2015) *Compromised Data: From Social Media to Big Data*, London: Bloomsbury.
 Fábos, A. and E. Haddad (2013) "Toward a Feminist Analysis of 'Impact': Sonda Hale's Scholarship and Activism in and Beyond the University," *Journal of Middle East Women's Studies* 10(1), 53–81.
 Fazi M. B. and Fuller, M. (2016) "Computational Aesthetics," in P. Christiane (ed.) *A Companion to Digital Art*, Oxford: Blackwell.
 Feuz, M., M. Fuller, and F. Stalder (2011) "Personal Web Searching in the Age of Semantic Capitalism: Diagnosing the Mechanisms of Personalization," *First Monday* 16(2).
 Fuller, M., N. Mazurov, and D. McQuillan (forthcoming) "The Author Field," in G. Bachmann, Y. Hui, and S. Lash (eds.) *Technics and Data*, London: Sage.
 Galloway, A. (2012) *The Interface Effect*, Cambridge, MA: Polity.
 Gehl, R. W. (2014) *Reverse Engineering Social Media: Software, Culture, and Political Economy in New Media Capitalism*, Philadelphia, PA: Temple University Press.
 Goriunova, O. (2010) *Funware* (Exhibition), Arnolfini, Bristol, September–November 2010
 Goriunova, O. (2011) *Fun with Software* (Exhibition), MU and Baltan, Eindhoven, November 2010–January 2011.
 Goriunova, O. (ed.) (2014) *Fun and Software: Exploring Pleasure, Paradox and Pain in Computing*, London: Bloomsbury.
 Grosser, B. (2014) Facebook Demetricator, retrieved from bengrosser.com/projects/facebook-demetricator.
 Helmond, A. (2013) "The Algorithmization of the Hyperlink," *Computational Culture* 3, retrieved from computationalculture.net/aerticle/the-algorithmization-of-the-hyperlink.
 Howse, M. (2015) "–micro research—Martin Howse [London/Berlin]," retrieved from 1010.co.uk/org.
 Kaldrack, I. and T. Röhle (2014) "Divide and Share: Taxonomies, Orders and Masses in Facebook's Open Graph," *Computational Culture* 4, retrieved from computationalculture.net/issue-four-2.
 Kemp, J. (2016) retrieved from xxn.org.uk.
 Kitchin, R. (2014a) *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*, London: Sage.
 Kitchin, R. (2014b) "Thinking Critically About and Researching Algorithms," *The Programmable City Working Paper* 5.
 Kopytoff, I. (1986) "The Cultural Biography of Things: Commoditization as Process," in A. Appadurai (ed.) *The Social Life of Things: Commodities in Cultural Perspective*, New York, NY: Cambridge University Press.
 Lash, S. (2002) "Technological Forms of Life," in *Critique of Information*, London: Sage.
 Mackenzie, A. (2008) "Codecs" in M. Fuller (ed.) *Software Studies*, Cambridge, MA: MIT Press.
 Mackenzie, A. (2010) *Wirelessness: Radical Empiricism in Network Cultures*, Cambridge, MA: MIT Press.
 Miyazaki, S. (2012) "Algorithimics: Understanding Micro-Temporality in Computational Cultures," *Computational Culture* 2, retrieved from computationalculture.net/article/algorithimics-understanding-micro-temporality-in-computational-cultures.
 Montfort, N., P. Baudoin, J. Bell, I. Bogost, J. Douglass, M. C. Marino, M. Mateas, C. Reas, M. Sample, and N. Vawter (2014) *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10*, Cambridge, MA: MIT Press.
 Parisi, L. (2013) *Contagious Architecture: Computation, Aesthetics and Space*, Cambridge, MA: MIT Press.

- Rancière, J. (1991) *The Ignorant Schoolmaster: Five Lessons in Intellectual Emancipation*, K. Ross (trans.) Stanford, CA: Stanford University Press.
- Sandvig, C., K. Hamilton, K. Karahalios, and C. Langbort (2014) "Auditing Algorithms: Research Methods for Detecting Discrimination on Internet Platforms," presented at *Data and Discrimination: Converting Critical Concerns into Productive Inquiry*, a pre-conference at 64th Annual Meeting of the International Communication Association, May 22, Seattle, WA.
- Skeggs, B. and S. Yuill (2015) *Values and Value*, retrieved from values.doc.gold.ac.uk/.
- Software Studies Initiative (2014) *Selfie City*, retrieved from www.selfiecity.net.
- Star, S. and J. Griesemer (1989) "Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907–1939," *Social Studies of Science* 19(3), 387.
- Sweeney, L. (2013) "Discrimination in Online Ad Delivery," *Communications of the AMC* 56(5), 44–54.
- Vee, A. (2012) "Text, Speech, Machine: Metaphors for Computer Code in the Law," *Computational Culture* 2, retrieved from computationalculture.net/article/text-speech-machine-metaphors-for-computer-code-in-the-law.
- Wilkie, A. (2010) "User Assemblages in Design: An Ethnographic Study," PhD thesis, University of London.
- Yokokoji M. and G. Harwood (n.d.) "YoHa (English translation 'aftermath')," retrieved from yoha.co.uk.