SOFTWARE

The term “software” harks back to the legal reordering of the computer industry in the 1950s and 1960s, when “digital objects, languages, and logical structures” were first perceived as having a common intangible nature and commercial potential (Fuller et al. 2008: 1). The interdisciplinary field of software studies investigates “the conditions of possibility that software establishes” in the broader media field (Fuller et al. 2008: 2). Practitioners in this area recognize affinities with work in computing history, science and technology studies, and software art. They extend to software diverse approaches in media studies (Manovich 2001; Lovink 2003; Galloway 2004) and anthropology (MacKenzie 2006), often interrogating software through both software writing and critical analysis.

A more expansive exploration of the political, historical and cultural significance of software and their effect on formal and/or conceptual dimensions of software would encompass an analysis of software as a key element in communications around the world. In this direction, historians (Marvin 1988; Adas 1989), scholars in science and technology studies (Bowker and Star 1999; Edwards et al. 2009) and anthropology (Larkin 2008) have emphasized that engagements around technological infrastructures are realms of simultaneous negotiation of
political, social, and cultural power. Other rich conceptual and empirical resources for understanding the cultural significance of software can be found in scholarship on software standards and law, which has clearly articulated that negotiations of property rights in information involve “contentious moral and political decision making about the distribution of wealth, power, and information” (Boyle 1996: 27; cf. DeNardis 2009; Lessig 1999; Goldsmith and Wu 2006). Complementing this research, anthropologists studying free software (Kelty 2008; Coleman 2009) and activists themselves (Aigrain 2005; Ghosh 2005) have argued that struggles over cultural meanings of software are central to shaping legal regimes, economic frameworks, and the perceived transformative potential of software.

Historians have traced multiple trajectories of software development in Cold-War military-industrial-academic research (Edwards 1997), countercultural communes (Turner 2006), and non-US contexts (Abbate 1999). By emphasizing contributions that tend to be obscured in contemporary debates about software (Light 1999), these perspectives expand what counts in understanding cultures of software and dovetail with growing scholarly interest in issues such as the training and career trajectories of software engineers (Margolis and Fischer 2002), as well as the reconfiguration of labor in the software industry (Biang 2006; Dibbell 2008; Malaby 2009).

References:


in the Late Nineteenth Century. New York: Oxford University Press.
