MAT LIT

Revista do Programa de Doutoramento «Estudos Avançados em Materialidades da Literatura»

New Media as Software

MANUEL PORTELA
CLP | University of Coimbra
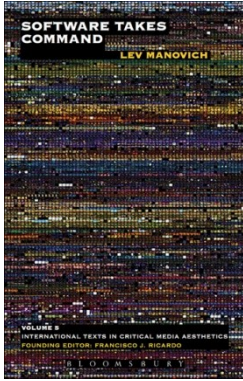
Lev Manovich, *Software Takes Command: Extending the Language of New Media.* London: Bloomsbury, 2013, 358 pp. ISBN 978-1-6235-6817-7.

July **2013 saw** the publication of the fifth volume in the series 'International Texts in Critical Media Aesthetics', directed by Francisco J. Ricardo. Following the volumes by C.T. Funkhouser (*New Directions in Digital Poetry*, 2012), Markku Eskelinen (*Cybertext Poetics: The Critical Landscape of New Media Literary Theory*, 2012; see review in MATLIT, Volume 1.1, pp. 208-212), Martha Buskirk (*Creative Enterprise: Contemporary Art between Museum and Marketplace*, 2012), and Francisco J. Ricardo (*The Aesthetic Engagement: Experiencing New Media Art through Critique*, 2013), *Software Takes Command* confirms the importance of this series in the renewal of critical thinking about digital mediation by its interdisciplinary method for addressing the technical, aesthetic, and social dimensions of ongoing processes. The development of a critical theory of media which attempts to be, at the same time, a critical aesthetics of media is a sign of the revaluation of the aesthetic and, in particular, of the appreciation of media arts in their multiple forms (digital art, digital music, digital literature) as practices and devices for interrogating the medial condition of human culture in the computer age.

In Lev Manovich's most recent book, this programmatic interrogation of our medial condition leads to the following question: *do media still exist after software?* This is the question that triggers Manovich's dialogue both with computing history and with theories of digital media of recent decades, including the extension of his own previous formulations in *The Language of New Media*, published in 2001, and which became a major reference work in the field. The subtitle of the new book points precisely to this critical

revisiting of his earlier work in the context of ubiquitous computing and accelerated transcoding of social, cultural and artistic practices by software. Its title, in turn, contains a reference to *Mechanization Takes Command* (1947), by Sigfried Giedion, a book that described the process of mechanization of society in many sectors of industry, commerce, and services. The book's title thus suggests a similar chain of multiple and wide-ranging effects based on the observation of the action of the digital computer as a tool for social reproduction.

This analysis of retroactivity between culture and software is described in three stages. The first part ('Inventing Media Software', pp. 53-157) describes the invention of media software based on a historical analysis of the conceptual models contained in the interfaces and programs designed by some of the engineers who theorized the features of interaction and visualization of the digital computer. Noteworthy inventors and thinkers discussed include J. C. R. Licklider, Ivan Sutherland, Ted Nelson, Douglas Engelbart and, in particular, Alan Kay, in whose ideas Manovich sees the foreshadowing of the 'universal media machine' that would come to define the shape of the digital computer as a set of applications for authoring and editing media objects in the same unified interface, currently embodied in the integrated operations of hardware, software, and network. The remediating nature of digital media, highlighted by Bolter and Grusin as one of their predominant formal principles, seems to have originated in a particular conceptual model for programming and interfacing. The second part ('Hybridization and Evolution', pp. 159-239) describes the strategies for hybridization of genres and forms that accompanied the transformation of the computer into a multimedia machine, which can be theorized either as a metamedium that remediates all other media, or as a monomedium that dissolves the boundaries of those media it absorbs. Finally, in the third part ('Software in Action', pp. 241-327), Manovich examines in detail various media software applications showing how the properties of software have become properties of media. Throughout the book, several applications are closely analyzed, namely, the image editing program Photoshop (pp. 124-147) and the video editing program After Effects (pp. 243-327), demonstrating how their layered compositional properties result in an aesthetics of hybridity, remixing, and variability.

One of the premises of Manovich is that software has become 'the engine of contemporary societies' to such a degree that the terms 'software society' and 'software culture' constitute by now appropriate metonymies to symbolize a wide range of processes of social reproduction, which extend to the economic, artistic and communication processes. Media software, the main object of *Software Takes Command*, would be one of those technocultural categories, through which it becomes possible to understand the over-determination of media content by software form. The two basic hypotheses posed by Manovich are that the specificity of new media culture can be

described from this software layer, and the materiality of digital processing, that is, the unification of multiple data streams via the same universal binary encoding process, calls into question the very separation of media as distinct technologies. The preponderance of the software as a common layer would imply the eventual dissolution of their technical, generic and formal identity.

An analysis of media archaeology shows that electronic media of late nineteenth century led to the replacement of direct inscription in a surface that was accessible to human senses (as was the case of engraving, letterpress, lithography and photography) by a set of electrical signals which had to be represented and controlled via an interface – for example, the frequency display and buttons of a radio device. The introduction of this interface for representation and control changed the operation of media, since their properties also came to depend on this interface. Digitization of media in the late twentieth century continued this separation of data display from their technical representation in the coding layer. With the representation of data in the form of numerical codes, these can only be accessed through software applications, thus instituting the separation between hardware and software. Properties of digital media are now determined by the specific properties of the software that makes them accessible and processable.

The history of computing shows that the development of programming and interfaces consisted essentially of exploring the simulating capabilities theorized by Alan Turing and John von Neumann when they conceptualized the computer as a multifunctional simulatory machine. Unlike the historical evolution of analog media towards their own particular languages, the digital computer seems realize itself ontologically through the simulation of previous media, constantly expanding their possibilities. This flexibility results from the separation of hardware from software, which makes possible a process of continuous experimentation that leads to the creation of the 'new' in new media, and which is described through an analogy with the avant-garde experimentation with processes and media:

> What differentiates a modern digital computer from any other machine – including industrial media machines for capturing and playing media – is separation of hardware and software. It is because an endless number of different programs performing different tasks can be written to run on the same type of machine, that that machine – i.e. a digital computer – is used so widely today. Consequently, the constant invention of new (and modification of existing) media software, is simply one example of this general principle. In its very structure computational media is "avant-garde" since it is constantly being extended and thus redefined. (92-93)

Extensibility and constant redefinition would result, ultimately, in the over-determination of the content of media by the software with which they are processed, i.e., produced, distributed, received, and appropriated. In the

context of a complete digitization of media, the postulate 'the medium is the message' can be rephrased as 'the software is the message'. In turn, the prominence of software as a common substrate for all media seems to imply the loss of operational value for the very notion of media inherited from previous technologies.

Other recent works, such as *Inventing the Medium* (2012) by Janet H. Murray, have proposed the replacement of the concept 'digital media' by 'digital medium', suggesting that the formal variability inherited from previous technologies is less decisive than their procedural identity. Friedrich Kittler had already described this unifying effect of machine-code and numerical representation when he invented a genealogy of media technologies as substitutes for the codes of writing: 'In computers everything becomes number: imageless, soundless, and wordless quantity. And if the optical fiber network reduces all formerly separate data flows to one standardized digital series of numbers, any medium can be translated into another. With numbers nothing is impossible. Modulation, transformation, synchronization; delay, memory, transposition; scrambling, scanning, mapping – a total connection of all media on a digital base erases the notion of the medium itself' (Kittler, *Literature, Media, Information Systems*, 1997, 31-32). Kittler radicalized the issue by describing the software itself as a mere epiphenomenon of the hardware, from whose electronic materiality all programmable and inscriptional forms ultimately depend (cf. Kittler, 'There is no software', 1995). Manovich responds provocatively with 'There is only software' (pp. 147-157), although his intention is similar, that is, to suggest that the concept of medium has to be rethought.

Other researchers, such as Wendy Hui Kyong Chun (*Programmed Visions*, 2011), have developed an analysis of the mystifying reification of software and, particularly, of so-called source-code, but without entirely denying the relative autonomy of programming languages at their various levels as specific sign systems. For Chun, software should also be analyzed as a discursive and cultural device. These are three distinct inflections that allow us to analyze the software at different observation scales, ranging from the relationship of circuits to machine code and to source code, and from the interaction of those codes to the general codes of culture and natural language. Manovich's approach is located halfway between kittlerian techno-determinism and chunian techno-discursive analysis, rehearsing a modulated reading of the interactions between software and culture based on the observation of the instrumental uses of software tools in manipulating media objects. The singularity of Manovich as a thinker of digital media comes precisely from his extraordinary ability for modular decomposition of the operations of the digital medium. His analysis of software layers combines a deep knowledge of the historical development of theories and computational technologies with a practical and poetic knowledge of the functionalities and algorithms of different programs for editing image, sound and video.

Software Takes Command accurately revisits and summarizes the history of human-computer interaction, while analyzing these interactions as they are embodied in our current software. The work develops its own critical language and provides a model for analyzing computer applications, theorizing them as formal components that are determinant in the creation of media and in the transcoding of cultural practices that incorporate media software. The intelligence and breadth of Manovich's approach makes this book relevant for all those who, in the fields of computer science, art, design, history and theory of new media, and related disciplines, wish to understand the multiple forms and structures of interaction and manipulation encoded in the software we use in many of our creative and communicative practices. Manovich beautifully synthesizes a significant part of his work as programmer, designer and digital animator, media artist, researcher and professor, helping to strengthen and expand the field of cultural studies of software, one of the corollaries of his previous systematic analysis of the language of new media. This is an essential book in the canon, still in formation, of software studies, and it should be added to books and essays by Matthew Fuller, Michael Mateas, Mark C. Marino, Noah Wardrip-Fruin, Nick Montfort and Wendy Hui Kyong Chun.