

Chapter 112

Software Literacy

Elaine Khoo

University of Waikato, New Zealand

Craig Hight

University of Newcastle, Australia

ABSTRACT

Software is not neutral. It comes with social and cultural assumptions that afford particular actions while constraining others. The notion of software literacy is emerging as one way to conceptualize the repertoires of skills and understandings needed for people to be critical and creative users of software packages and systems in a software saturated culture. This conceptual model is a response to current digital literacy frameworks which do not identify the implications of the choice of software on what can be achieved. Studies on information literacy and on ways of mastering software have tended to ignore the role of software itself. The study of software is only now emerging as a field of study. This contribution argues for the relevancy of software literacy as part of understanding the ways people engage with software and how its affordances influences knowledge representation, generation, and critique. It will define the term and set out three progressive tiers of development towards software literacy.

INTRODUCTION

Software mediates almost every aspect of everyday life. Nearly all of human professional and personal activities are embedded and shaped within systems and interactions that involve software at some level from the seductive ecosystems on devices such as iPhones or iPads through to functional interfaces of automatic teller machines (ATMs), to the tools underlying everyday practices such as word processing and email, through to the sophisticated professional editors for multimedia design. None of these tools are ‘neutral’. They derive from social and cultural assumptions about their use and all have particular values embedded in their interfaces and hierarchy of affordances. These embody conceptual frameworks in which particular ways of acting or thinking are more possible and imaginable than others. Despite their ubiquity within contemporary society, software itself is only now emerging as a field of study in its own right. As championed by the proponents of Software Studies this is a vital but neglected area of cultural production intersecting with and potentially shaping all other spheres of cultural, economic

DOI: 10.4018/978-1-5225-7598-6.ch112

and political activity (Fuller, 2008; Livingstone et al. 2014; Manovich, 2008). This article proposes the notion of ‘software literacy’ as one way to understand the skills and understandings needed for people to be critical and creative users of software (as application, platform and infrastructure) in today’s software saturated culture. This contribution argues for the relevancy of software literacy as deeply intertwined with people’s engagement with software and how it influences the way people come to understand, represent, generate and critique knowledge. The discussion below begins by overviewing the background and need for a focus on software studies before proceeding to define, introduce and elaborate on a framework for software literacy. An outline of the framework is then grounded in and exemplified through a case study located within a university teaching and learning context with implications for further thinking and research in the field.

BACKGROUND

The proliferation of digital and networked technologies is an expanding and accelerating feature of modern societies and can be predicted to continue to rise and impact on almost every sphere of human living. Most people develop some level of proficiency with everyday software packages informally through their daily use and incremental engagement over time (Bulfin & Koutsogiannis, 2012; Hague & Logan, 2009). Informal learning practices have been shown to increase students’ sense of agency and consequently to have the potential to make learning a richer and more fulfilling experience (Furlong & Davies, 2012). Commenting on the trend of digital penetration, numerous authors have further argued that ubiquitous access to digital technologies has shaped a new internet-centred generation of ‘digital natives’ (Oblinger, 2003) with the corresponding assumption that access to digital tools has, on its own, facilitated the development of new learning skill sets (Tapscott, 2009). Terms such as the ‘digital generation’, ‘millennials’, ‘Net Generation’ (Tapscott, 1999), ‘digital natives’ (Prensky, 2001), ‘Google generation’, ‘Generation Y’ and so forth have derived from a host of assumption about the distinctive skill set of generations immersed within digital technologies. Such labels aim to characterise an emerging class of learners accustomed to engaging with software and technologies such that they can effortlessly adopt new technologies, operate at ‘twitch speed’, are able to multitask, imagine, and visualize while communicating in multiple modalities and consequently possess higher technical skills compared to previous generations (Prensky, 2001). The term ‘digital natives’ itself assumes a generational change in digital literacies fed particularly by informal learning, opinions which are closely informed by emancipatory rhetoric surrounding the digital. Consequently in part the term articulates anxieties amongst educational institutions and practitioners that they are falling behind the literacies students will bring with them to learning contexts.

Educators thus are often encouraged to operate under the assumptions that students already possess the necessary computing skills and conceptual frameworks to learn with and through generic software packages, and tend to neglect the role which the affordances of software themselves play in shaping how students ‘perform’ the software (Adams, 2006). Recent research indicates that such assumptions about students’ digital proficiencies are unfounded and that digital inequalities and marginalization persist around students’ access to, and use of, information and knowledge (Bennett, Maton, & Kervin, 2008; Jones & Czerniewicz, 2010; Kennedy et al., 2008). There is evidence students may not be aware of how to apply software embedded technologies effectively to enhance their learning (Khoo, Johnson, & Zahra, 2012; Valtonen, et al., 2011). In other words, digital inequality may not be defined specifically to the issue of

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/software-literacy/214718

Related Content

Improving Throughput of Starved TCP Flow by Sidestepping Bottleneck Nodes Using Concurrent Transmission

Rajesh Verma, Arun Prakash, Rajeev Tripathi and Neeraj Tyagi (2010). *International Journal of Mobile Computing and Multimedia Communications* (pp. 68-83).

www.irma-international.org/article/improving-throughput-starved-tcp-flow/40981

Business and Technology Issues in Wireless Networking

D. Wright (2007). *Encyclopedia of Mobile Computing and Commerce* (pp. 90-95).

www.irma-international.org/chapter/business-technology-issues-wireless-networking/17058

Survivability in RFID Systems

YanJun Zuo (2010). *Handheld Computing for Mobile Commerce: Applications, Concepts and Technologies* (pp. 300-312).

www.irma-international.org/chapter/survivability-rfid-systems/41638

A Pricing Model for Effective Radio Spectrum Utilization

Sunil Kumar Singh and Deo Prakash Vidyarthi (2019). *International Journal of Mobile Computing and Multimedia Communications* (pp. 41-65).

www.irma-international.org/article/a-pricing-model-for-effective-radio-spectrum-utilization/241787

Advanced Resource Discovery Protocol for Semantic-Enabled M-Commerce

Michele Ruta, Tommaso Di Noia, Eugenio Di Sciascio, Francesco Maria Donini and Giacomo Piscitelli (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications* (pp. 2957-2968).

www.irma-international.org/chapter/advanced-resource-discovery-protocol-semantic/26704