Thinking Skills and Creativity 22 (2016) 97–110

Contents lists available at ScienceDirect

Thinking Skills and Creativity

journal homepage: http://www.elsevier.com/locate/tsc

Metaphors of code—Structuring and broadening the discussion on teaching children to code

Tomi Dufva[a,*], Mikko Dufva[b]

[a] Aalto-University, School of Arts, Design and Architecture, Finland
[b] VTT Technical Research Centre of Finland Ltd, Tekniikankatu 1, Tampere, P.O. Box 1300, 33101 Tampere, Finland

ARTICLE INFO

Article history:
Received 29 December 2015
Received in revised form 22 August 2016
Accepted 7 September 2016
Available online 15 September 2016

Keywords:
Code
Code literacy
Metaphors
Education
Programming
Teaching programming
Pedagogy
Media literacy

ABSTRACT

Digital technology has become embedded into our daily lives. Code is at the heart of this technology. The way code is perceived influences the way our everyday interaction with digital technologies is perceived: is it an objective exchange of ones and zeros, or a value-laden power struggle between white male programmers and those who think they are users, when they are, in fact, the product being sold. Understanding the nature of code thus enables the imagination and exploration of the present state and alternative future developments of digital technologies. A wider imagination is especially important for developing basic education so that it provides the capabilities for coping with these developments. Currently, the discussion has been mainly on the technical details of code. We study how to broaden this narrow view in order to support the design of more comprehensive and future-proof education around code and coding. We approach the concept of code through nine different metaphors from the existing literature on systems thinking and organisational studies. The metaphors we use are machine, organism, brain, flux and transformation, culture, political system, psychic prison, instrument of domination and carnival. We describe their epistemological backgrounds and give examples of how code is perceived through each of them. We then use the metaphors in order to suggest different complementary ways that ICT could be taught in schools. The metaphors illustrate different contexts and help to interpret the discussions related to developments in digital technologies such as free software movement, democratization of information and internet of things. They also help to identify the dominant views and the tensions between the views. We propose that the systematic use of metaphors described in this paper would be a useful tool for broadening and structuring the dialogue about teaching children to code.
© 2016 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Digitality as a phenomenon defines our era. Digital technologies have secured their place in business and in social relations as well as in culture. Digital technologies affect society, but often these changes are taken as given, without broader discussion on the impacts and consequences (König et al., 1985). This is troubling, because digital technology functions in various positions in our society. For example, a high percentage of stock trading is done through trading algorithms with little human involvement. (Washington, 2015; Steiner, 2013). Modern cars carry so much digital technology they have been called "computers on wheels" (Foley Lardner LLP, 2014; Hirsch, 2015). Social media, essentially a digital phenomenon, has defined

* Corresponding author.
E-mail addresses: tomi.dufva@aalto.fi (T. Dufva), mikko.dufva@vtt.fi (M. Dufva).

http://dx.doi.org/10.1016/j.tsc.2016.09.004
1871-1871/© 2016 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

new ways of interaction and has influenced culture. There is also evidence that digital technologies shape the way people think, by supporting, sharing and expanding people's cognitive processes(Barzilai and Zohar, 2006). By digital technologies, we mean technologies that are based on digital signal processing, which can be reduced to a flow of ones and zeroes, and which usually utilize information networks to function.Digital technologies allowed for the rampant innovation and growth that started around the 1940s and are defined as the digital age (Ceruzzi, 2012). Digital technologies include all the technologies from smartphones and computers to automated manufacturing and decentralized communication protocols. Digitalization presents new challenges, that, in essence, call for an understanding of digital technologies. The so-called digital divide, that formerly implied the distinction between those who have access to the internet and to those who do not (Mehra, Merkel, & Bishop, 2004) can now be seen as the divide between those who understand digital technologies and those who do not. (For a historical view on ICT in education, see Wilson, Scalise, & Gochyyev, 2015). Mark Warschauer points out that, in today's society, the ability to access, adapt and create knowledge using information and communication technologies is critical to social inclusion (Warschauer, 2004).

The access to digital resources, as well as the ease of use of those resources, has increased, but the understanding of the code has not kept the same pace. This can be seen, for example, within the digital natives discussion. Knowing how to use a tablet computer at the age of two does not mean that one understands the way the machine works or the code behind it. It does not even imply that one could learn to cope with the technology (Kupiainen, 2013). This can also be seen from Carita Kiili's dissertation (Kiili, 2012) where she states that many young adults have problems assessing and evaluating search results in the net. In essence, digital technologies are a source of inequality, which is problematic given their ubiquity in modern society.

Code is the heart of every digital technology and substantially shapes its behaviour. In this paper, we define code as a digital language with a set of assumptions about the users and the world. Code is used to create programs that control digital technologies, from automated factories to personal computers, and from connected home appliances to services providing social networking. Thus, code, in our working definition, refers to the principles and choices made, and is not restricted to any specific programming language. Coding is the act of writing code and building programs, which includes making implicit and explicit choices about the purpose, framing and scope of the program.

The key motivation for this paper is that, because digital technologies are always programmed and are thus based on code, understanding code and the assumptions inherent in it is necessary for full participation in modern society. The code in digital technologies is not value-free, rather it widely reflects both conscious and subliminal values of the programmer, a software company or society's understanding of good code. Digital technology's operating models are not immutable laws of nature, but rather flexible models that are designed and controlled by humans (Lessig, 1999, 2009). Code does not reflect objective truth about the world. Instead, it constructs laws in the digital realm. Without understanding how these laws are formed, we are not able to fully participate in the discourse of our digital life (Giroux, 2011; Lessig, 2009, Rushkoff, 2010). Technology does not impinge upon us from the outside of society, but interweaves into our society in the same way as the political or economic system does, and is also dependent on these other systems, which can alter the way, or speed, of technological progress (König et al., 1985). Without including technology as a coherent part of societal discussion the effects of technology and its relations to other systems stay ambiguous. Furthermore discussion around the ramifications of technologies are crucial as technology has the tendency to convert social, scientific, governmental and human problems into technical problems (Williamson, 2015).

We propose code literacy as a way to participate to the discussion around the effects of digital technologies on society. Code literacy does not directly allude to learning to program in the traditional sense, rather it implies the understanding of the code and its intentions and context. The notion of literacy illustrates the case: In the same way that not all literate individuals become authors, not all code-literate individuals become developers. Still, literate people have the necessary skills and the apprehension of reading and writing.

Understanding code does not emerge naturally from lived experience, but has to be taught. The code used to form the present digital world, be it an operating system, software or stock- trading algorithm, is distinctly different from the everyday analogue tools, such as hammer, pen or paintbrush, used to form the material world. One example of this is the binary system of two alternate states, often represented as 1 and 0. Code is binary and, therefore, can be reduced to "yes or no" decisions. However, as Rushkoff argues, human lives are not binary and thus trying to represent them using these binary systems is problematic (Rushkoff, 2010).

Learning to code and digital learning systems are deeply intertwined in political, societal and commercial structures (Williamson, 2015, 2016). We argue that current teaching about digital technologies, programming and code and the discussion around it does not take fully into account the societal and ethical dimensions of code. Thus, our goal in this paper is to broaden the discussion and propose a structure for understanding different views on code. To facilitate this, we describe nine metaphors of code based on four paradigms. Through the use of metaphors and their associated paradigms we wish to support a larger and more holistic view on code and digital technologies.

This paper is structured as follows. After this introduction, in Section 2 we describe nine general metaphors that cover four common paradigms of social theory as well as different assumptions about the complexity of the world and the relations between stakeholders. In Section 3, we apply these metaphors to structuring the discussion around code and illustrating various viewpoints expressed about what code is and how it influences society. In Section 4, we focus specifically on education around code and coding, and suggest different views on teaching code. Section 5 concludes the paper.

Download English Version:

https://daneshyari.com/en/article/4941896

Download Persian Version:

https://daneshyari.com/article/4941896

Daneshyari.com