Chapter XIX

The Philosophies of Software

Lindsay Grace
University of Illinois and Illinois Institute of Art, USA

ABSTRACT

Software is philosophical. Software is designed by people who have been influenced by a specific understanding of the way objects, people and systems work. These concepts are then transferred to the user, who manipulates that software within the parameters set by the software designer. The use of these rules by the designer reinforces an understanding of the world that is supported by the software they use. The designer then produces works that mimic these same philosophies instead of departing from them. The three axes of these philosophies are analogy, reductivism, and transferred agency. The effects on computer-based artistic expression, the training in digital art production, and the critique of art are evaluated in this chapter. Tensions between the dominant scientific approaches and the dominant artistic approaches are also defined as destructive and constructive practice respectively. The conclusion is a new critical perspective through which one may evaluate computer integrated creative practice and inspire fresh creative composition.

INTRODUCTION

There is a simple logical proof that describes software’s relationship to philosophy. Software is designed. Design prescribes philosophies. Since software is designed, it must also dictate philosophy. The existence of these philosophies, their sociological effects, and the need to critique these philosophies is the focus of this writing. This writing does not seek to define ontologies of philosophies, nor does it seek to be an exhaustive examination of the many philosophies that have been institutionalized into the practices of developing and using software. Instead, this article seeks to highlight the existence of a few important philosophies in an effort to encourage practitioners to critically examine their relationship to software and its effects on their practice. In particular, critical assessment of software philosophies engenders fresh approaches to universal, original and effective design.
There are several existing areas where philosophy exerts an influence on software. Each of these areas is not only affected by inherent philosophies, but each area inspires the growth of their individual philosophies by the design and use of their systems. In some cases, the philosophy intersects to create a fulcrum on which multiple assumptions about the construction of the world express themselves. The following sections attempt to outline a few of the major philosophical undertones of common software applications as they relate to the Design of User Interfaces, Avatars, and the use of object orientation.

Careful examination of software decants the following key philosophical elements:

- The heavy use of analogy
- The application of reductivism
- An emphasis on transferred agency

Each of these elements directs users toward specific modes of operation, problem solving and creative efforts. This chapter concerns itself with the identification and evaluation of the philosophies resulting from the use, either successful or unsuccessful, of software built with these elements. The final section of this writing highlights how these philosophies instruct software users.

**BACKGROUND**

For some, Philosophy is a term that should not be paired with software. Within this subset, philosophy is abstract, whereas computer software design is science. Granted that there are scientific underpinnings to software, it is important to recognize that software is used in increasingly abstract ways. It is used to create art, it is used to communicate, and it is used as an integral part of daily work that involves abstract thinking.

The philosophy of software is a topic of research and rhetoric in many disciplines. Although not always considered a philosophical examination, practitioners of law, education, commerce and nearly every software-effected discipline have discussed a kind of philosophy of software. These concerns include intellectual property rights, electronic learning, and the design of systems. The philosophical and commercial work of the Free Software Foundation, for example, is directed toward the specific effect software production philosophies have on the quality of software produced. Theirs’ is largely an examination of how software production is practiced, not an examination of how software effects production. This writing seeks to expose the effects of philosophies so ingrained in the production of software that they are seemingly transparent. In the oft-used paraphrase of Marshal McLuhan, we shape our tools and then our tools shape us (1994).

It is important to note that this discussion excludes an examination of hardware’s role. This is because hardware finds design from the realities of physical sciences, where software finds design from logic. The critical evaluation of this logic decants priorities, ideologies, and value systems. Simply stated, it influences the foundations of existing philosophies. Those philosophies are encoded in the language and structure of software, and interpreted by the user.

Investigations into the effect of language on people’s ability to understand specific ideas are perhaps more akin to the focus of this chapter. In Noam Chomsky’s (2006) essays, (compiled later in the book Language and Mind), he encouraged critical assessment of the relationship between linguistics and philosophy that opened for examination whole processes of communication and approach. The role of linguistics as a tool through which we produce and communicate meaning is similar to the role of software. Software serves as the tool through which much daily production and communication occurs, making it the lingua franca of operation in most of the western world. It is the basis for communication and the primary vessel that facilitates communication. Programming languages are also the tools we use to direct the development of a solution.
Related Content

Effects of Editing Style on the Perceived Meaning: A Comparative Study Between Old and Young Egyptian Television Audiences
[www.igi-global.com/article/effects-of-editing-style-on-the-perceived-meaning/90152?camid=4v1a](www.igi-global.com/article/effects-of-editing-style-on-the-perceived-meaning/90152?camid=4v1a)

Design for Consciousness in the Wild: Notes on Cognition and Space
[www.igi-global.com/chapter/design-for-consciousness-in-the-wild/138547?camid=4v1a](www.igi-global.com/chapter/design-for-consciousness-in-the-wild/138547?camid=4v1a)

Virtual Reality and Learning in an African University Environment: Trends and Contextual Issues
[www.igi-global.com/article/virtual-reality-learning-african-university/54236?camid=4v1a](www.igi-global.com/article/virtual-reality-learning-african-university/54236?camid=4v1a)

Intercultural Dialogue through Design (iDiDe): A Model of Intercultural Collaboration and Student Engagement
Susan Ang (2017). *Collaboration and Student Engagement in Design Education* (pp. 230-256).
[www.igi-global.com/chapter/intercultural-dialogue-through-design-idide/165684?camid=4v1a](www.igi-global.com/chapter/intercultural-dialogue-through-design-idide/165684?camid=4v1a)