Chapter 5
On the Origins and Basic Aspects of User-Centered Design and User Experience

Rodrigo Hernández-Ramírez

Abstract Over a decade after the iPhone was first commercialized, screen-based tactile interfaces have become our primary means for interacting with computational technologies; although thanks to recent developments in Machine Learning (ML), gesture and voice control mechanisms will become more common. Smart technologies and what we end up defining and recognizing as robots will determine the practical principles, and the type of experience designers will be able to shape. To understand where the future of User-Centered Design and User Experience will take us, we need to understand how we got to where we are. The main goal of this chapter is clarifying what may be understood by User-Centered Design and User Experience. To do so, it will look at the period between the late 1960s and early 1980s, when Personal Computing and the ensuing democratization of technology forced designers to think about their users honestly.

Keywords Aesthetics · Artificial agents · Human–computer interaction · Human-centered design · Pragmatism · Technology · User experience · User experience design

5.1 Introduction

Over the last four decades, computers went from being rare tools for specialists to ubiquitous personal, intimate devices.[1] Nowadays, computational technology has already been—or can be potentially—integrated into every other artifact, including cars, telephones, vacuum cleaners, stoves, thermostats, or toilets. Thanks to high-speed data transfer, the increasingly potent sensors, and processors embedded within

[1] As Kay (2002, p. 124) notes, when he first began to conceive the "personal" computer, the idea he had in mind was "intimacy".

R. Hernández-Ramírez (✉)
Universidade Europeia, IADE, Av. D. Carlos I, 4, 1200-649 Lisbon, Portugal
e-mail: rodrigo.ramirez@universidadeeuropeia.pt

UNIDCOM/IADE – Unidade de Investigação em Design e Comunicação, Av. D. Carlos I, 4, 1200-649 Lisbon, Portugal

H. Ayanoğlu and E. Duarte (eds.), *Emotional Design in Human-Robot Interaction*, Human–Computer Interaction Series, https://doi.org/10.1007/978-3-319-96722-6_5

these artifacts can gather and receive vast amounts of data. Some of these datasets, in turn, are used alongside new algorithmic methods to train so-called "smart" software systems. These Machine Learning methods are behind the third "boom" or "wave" (Garvey 2018; Lee 2018) of Artificial Intelligence (AI), which can take raw visual and auditive information such as images and voice directly as input. This type of AI is powering a new generation of smart appliances that we would normally refer to as artificial agents (AAs) or robots.[2] Paradigmatic examples of the latter include autonomous vehicles, the highly publicized robotic humanoids, and quadrupeds developed by Boston Dynamics, and AI-powered personal assistants[3] such as Apple's "Siri", Microsoft's "Cortana", or Google's Assistant.

These devices have increasing agency[4] and autonomy[5] thanks to ubiquitous computing, which has "enveloped"[6] our environment, making it more accommodating for AAs (Floridi 2014). Our world, which until recently harbored only "dead" objects, will be further populated by responsive, interactive, and interconnected systems. As this Internet of Things (IoT) trend continues, more aspects of concrete reality will be incorporated into our informational environment or "infosphere" (see Floridi 2002). Consequently, hitherto meaningful boundaries between the (linear, Newtonian, and inert) offline world and the (post historical, informational, alive) online world will become irrelevant. Our interaction with information technologies (ITs) will stop being primarily mediated by screens and instead will be embedded in our environment (Lee 2018). This situation brings a host of issues, but also possibilities for Interaction Design (IxD) and User Experience (UX) at large, not to mention that some of the principles behind User-Centered Design (UCD) will need to be revised and updated to meet the challenges of new Human–AA relationships better.

Both UCD and UX are tied to Donald A. Norman and the work he developed as an academic and as a design consultant working for major technology companies. Many of the principles and methods comprising UCD can be traced back to Gould's and Lewis (1985) article "Designing for Usability" and the Project on Human–Machine Interaction from the Institute for Cognitive Science at the University of California in which Norman played a central role (see Norman and Draper 1986). In terms of design

[2] Paraphrasing Bryson (2009, 2019), a robot is any nonhuman artificial agent that transforms perception into action; any artifact that senses and acts in the physical world, and in real time. Thus, under this broad definition, smartphones, smart speakers such as Amazon's Echo ("Alexa"), and even "Roomba" vacuum cleaners count as robots, but so do disembodied AAs such as chatbots. In this chapter, the terms "robot" and "artificial agent" will be used indistinctly.

[3] According to Danaher (2018), a "personal AI assistant" is any software system that can act autonomously in a goal-directed manner. That is to say, any software that can receive instructions (usually by voice) as input and provide specific output, such as directions to a place, selecting among a range of options.

[4] Understood as the capacity to decide to act (or not), to choose the means to do so, and to apply those means to bring about specific changes in the world (Bunnin and Yu 2009, p. 19).

[5] In the context of AI, "autonomy" has a distinctive, more restrictive, meaning of a system being capable of achieving a given goal without having their course of action fully specified by a human (Johnson and Verdicchio 2017).

[6] An envelope or "reach envelop" refers to the three-dimensional bounded space in which a robot can reach (Floridi 2013).

practice, there is a growing number of methodologies based on UCD principles. Some of the most influential approaches are Norman's (2013) own Human-Centered Design (HCD) and Goal-Directed Design by Cooper et al. (2014).[7] Whereas more recent iterations include Wright's and McCarthy's (2010) Experience-Centered Design, Karjaluoto's (2013) "knowledge-led, systems based" Design Method, and perhaps the most well-known of all, IDEO's Design Thinking[8] (Brown 2008, 2009; Meinel et al. 2011).

The term UX was allegedly invented by Norman in the early 1990s while working at Apple's Advanced Technology Group (Buley 2013; Merholz 2007; Norman et al. 1995). Norman contended that "user experience" would better characterize an expanding area of design practice that could no longer be described through interface design and usability (Merholz 2007). Hassenzahl's (2013) entry in the *Encyclopedia of Human-Computer Interaction*, as well as his book, "Experience Design" (2010), provide thorough descriptions of UX principles. Wright and McCarthy (2010), and McCarthy and Wright (2004) also give a good account of UX design focusing on its philosophical and aesthetic roots. Besides providing a host of practical insights Saffer's book, *Designing for Interaction* (2010) contains a widely distributed cartography of the various disciplines associated with UX. UCD and UX are the two dominant paradigms in contemporary design practice. Nonetheless, a decade after popularizing the term, Norman lamented that UX and HCD, along with usability and "even affordances" have turned into buzzwords (see Merholz 2007). That people now use these terms with little or no awareness of their, origin, history, and the actual meaning.

This chapter takes the circumstances previously discussed as a starting point. Its main goal is clarifying what may be understood by UCD and UX. Its primary assumption is that to understand how these concepts will change as nonhuman agents further populate our environment, we first need to understand their origins. To do so, we need to look at the period between the late 1960s and early 1980s, when technical means and the cultural environment converged to allow the crystallization of the Personal Computer (PC) and the emergence of contemporary HCI.

5.2 Before There Were UCD and UX

Nowadays, the idea that computers are personal and even *intimate* devices are taken for granted, and so is the fact that we interact with them primarily through Graphical

[7] Cooper et al. (2014, pp. 13–14) make a point of distinguishing their approach from Norman's, which they claim is heavily inspired in Activity Theory, a conceptual framework initially developed by Russian psychologist Aleksei Leontiev (see Kaptelinin 2013; see also Kaptelinin and Nardi 2012).

[8] It is essential to distinguish between design thinking as an epistemic domain halfway between logical rationality and artistic intuition—i.e., as understood by Archer (1979) and Cross (2011)—and Kelley's and Brown's five-step general methodology for problem-solving (see Hernández-Ramírez 2018).

User Interfaces (GUI). This was not always the case. Before the 1970s, computational technology was not accessible to everyone, instead it was rigidly controlled by the government, educational, and private institutions. Even within universities, access to computers outside certain institutes was only possible through time-sharing,[9] which could be quite expensive.[10] Interacting with computers required at least basic knowledge of programming since the only way to work with them/issue commands was by inputting text through a terminal.

Since the mid-1950s, the United States Military (mainly through DARPA) had been financing research to improve the usability of computers. The public and private institutions doing it mostly followed ergonomic principles. Hence, as a distinctive field of research, Human-Computer Interaction (HCI) only emerged in the 1980s, around the same time computers became PCs, that is, consumer products for the general population (Carroll 2013). HCI marked a profound shift in the way engineers who had been developing computational technology in the last decades regarded end-users: they realized that nonspecialists had "functioning minds" and that understanding those minds would determine the way people would relate to computers in the future (Kay 2002). This shift had been gestating since the late 60s but arguably only turned paradigmatic once all the necessary components that are now familiar in every computer came together: microprocessors, pointing devices, and the GUI.

The icons and graphical representations comprising the GUI enabled every potential user to conceptualize computational process in more familiar terms through visual metaphors of "real life" objects and actions. While pointing devices allowed them to interact with computers more intuitively, by selecting and "touching" objects in the screen. The principles behind the GUI were based on intuitive learning and creativity; they contemplated and made explicit a factor which had hitherto been absent from interface design: the aesthetic dimension. To understand the origins of HCI and later, of UCD and UX, we need to make a short digression to look at the origins of the GUI and the notion of personal computing in general.

5.3 Early HCI and Interface Design

The origins of the GUI can be traced back to Ivan Sutherland's "Sketchpad" (1964), a computer program he developed during his Ph.D. that would revolutionize HCI, computer graphics, and the very notion of the computer. The primary goal of Sketchpad was allowing users to generate graphics not by writing code but by directly "drawing" over the monitor with a light pen. With Sketchpad, Sutherland introduced a new paradigm of interactivity, wherein by manipulating an image displayed on

[9] Time-sharing allowed multiple users to work on the same computer using different terminals, receiving alternated "slices" of computer execution time as they became available (Alesso and Smith 2008, p. 61).

[10] One hour of time-sharing could cost between 10 and 20 USD (Campbell-Kelly et al. 2014), that is, between 50 and 100 USD in today's money.

the screen, a person could directly change "something in the computer's memory" (Manovich 2002, p. 104).

Among the people influenced by Sketchpad was Douglas Engelbart. Engelbart was the founder of the Augmentation Research Center at the Stanford Research Institute. Inspired by Bush's (1945) seminal article, "As We May Think",[11] Engelbart had been attempting since the mid-1950s to develop a computer-based "personal information storage and retrieval machine" (Campbell-Kelly et al. 2014, p. 258). In late 1962, Engelbart obtained funding to develop what he and his research team called the "electronic office", a computer system capable of integrating for the first-time text and pictures (2014, pp. 258–259). Five years later, Engelbart's group was already prototyping what would arguably become their most lasting contribution to HCI: the computer mouse. After extensive testing, this peripheral showed to be more effective than the "light pen" used by Sketchnote and other joystick-like devices (Ceruzzi 2003). On December 9, 1968, at the Fall Joint Computer Conference in San Francisco,[12] Engelbart and about a dozen other people—including Stewart Brand, editor of the highly influential zine *The Whole Earth Catalog*—staged what came to be known as "The mother of all demos". Using a video projector to enlarge a computer screen to six meters, Engelbart showed the mouse, hypermedia, and teleconferencing; all of the features that would end up defining the contemporary computing environment.

Engelbart's electronic office system was too expensive to be commercialized due to a lack of cost-effective technology,[13] but the demo made a profound impression on the emerging HCI research community. Engelbart and his group conceived the feasible technological means for interacting with the computer beyond inputting text with a keyboard. But it was a group of researchers from the University of Utah—where Sutherland was a professor at that time—who conceived the software, and the visual language that eventually allowed computers to become personal tools. And arguably the most influential of them was Alan Kay.

As a doctoral candidate at the University of Utah, Kay pursued an ambitious project that would culminate in his thesis, *The Reactive Engine* (1969). In the thesis, he specified a new programming language called FLEX, as well as an early prototype for a personal computer that he designed along Ed Cheadle. According to Kay, the computer used a pointing device, a high-resolution display for text and animated graphics, and used the concept of multiple windows, but the interface, nonetheless, "repelled end-users" (2002, p. 123).

In 1972, Kay joined the recently founded Xerox Palo Alto Research Center (Xerox PARC) along with many of Engelbart's former colleagues (Ceruzzi 2003). This laboratory would be responsible for developing the Ethernet, laser printing, Object-Oriented programming, as well as the concept of the contemporary personal

[11] See Engelbart (1962) for his account of Bush's influence.

[12] For a full account, see "A research center for augmenting human intellect" (Engelbart and English 1968).

[13] At that time, even so-called "mini-computers" would cost several thousand dollars. The Intel 4004 microprocessor, which powered the new generation of PCs, only entered the market in 1971.

computer. By 1973, Kay and his team had developed a prototype computer called the “Xerox Alto”, whose operating system and configuration owed considerably to FLEX. The Alto was a desktop machine, it had a custom-built bitmap screen roughly equivalent to a letter-sized sheet of paper (215.9 by 279.4 mm) but oriented in portrait instead of landscape mode. The alto displayed documents that “look[ed] like typeset pages incorporating graphical images” (Campbell-Kelly et al. 2014, p. 260), and each one of the visible elements on it could be manipulated. Users could “scale letters and mix text and graphics on the screen” (Ceruzzi 2003, p. 262), which meant editing was effectively “what-you-see-is-what-you-get” (WYSIWYG). Having refined Engelbart’s design, Kay and his team incorporated the mouse into the alto, along with the “now-familiar desktop environment of icons, folders, and documents” (2014, p. 260). However, the Alto was never commercialized; at 18,000 USD a piece — about 90,000 USD in today’s money (Ceruzzi 2003, p. 261)—it was simply too expensive.

In 1979, Steve Jobs visited the Xerox PARC and was so impressed by the Alto that he convinced his partners (Steve Wozniak and Ronald Wayne) to incorporate the GUI paradigm into Apple computers. According to Kay’s account (2017a) Jobs was so amazed by the GUI that he missed the fact that the Alto had already incorporated networking (ethernet) and Object-Oriented Programming, two features that are indispensable in contemporary systems.

In 1981, Xerox introduced a commercial version of the Alto, the “Xerox 8010 Star System”, which was targeted at business users. Besides having a mouse and network connection, it was the first commercial computer to use a GUI based on the office “desktop” metaphor simulating interactable objects such as documents, folders, trash bin, rulers, pencils, “in” and “out” boxes, etc. (Brey 2008). The operating system allowed the user to treat *everything* that was displayed on the monitor (images, characters, words, sentences, paragraphs) as “objects” and thus select and manipulate them individually. Object integration was system-wide so that a document could hold charts, tables, and image modules along with the text. Moreover, the system incorporated generic commands (such as move, copy, open, delete, and show properties) that could be used on *every* object selected, using dedicated keyboard buttons. These features liberated the user from having to remember specific commands (e.g., Ctrl + C) to apply changes (Johnson et al. 1989).

The Xerox Star was conceptually and technically superior to every other office machine available at the time, but it was a commercial failure (Campbell-Kelly et al. 2014; Ceruzzi 2003). It was too expensive (it sold for approximately 16,500 USD), almost five times the price of other computers available at the time (Johnson et al. 1989; Smith and Alexander 1988). Furthermore, to take advantage of the Star’s distributed (Ethernet-based) networking, the potential buyer had to acquire at least two or three workstations along with a file server and one or two laser printers. That meant spending between fifty and a hundred thousand USD (almost a quarter of a million USD in today’s money). But the other major obstacle the Star faced was *conceptual*, and those responsible were Xerox’s salespeople as well as the potential buyers themselves.

5.4 The Computer Becomes Personal

As previously noted, before the mid-1970s, the very idea of a personal computer was not mainstream. The Star was advertised depicting an executive making calls, writing, and sending documents while sitting at his desk. Somehow the marketing department at Xerox failed to see that in those days' executives rarely, if ever, carried out any of those tasks (Ceruzzi 2003, p. 263). And even if a technologically curious executive would be willing to try a computer, he or she could buy and experiment with a far cheaper one (Smith and Alexander 1988). In contrast to Xerox's strategy, other brands (such as the now-defunct Wang Laboratories) aimed their products precisely at the people whose work conditions could be improved by using a PC: secretaries and office clerks. By then, the PC had been defined physically as an artifact, conceptually, however, it remained unclear why anyone would be interested in having one at home or work. The cultural environment was not yet ready for advanced personal information systems.

At that time, computers were still regarded as single-task devices meant for institutions. While in theory, the computer is a universal machine (Turing 1937), in practice mainframe and "mini" computers were fixed, and their reprogramming required specialized knowledge and hardware adjustments. For that reason, large companies such as IBM not only sold (or rather leased) computers but also "business services". Mainframe computers were custom-designed and programmed to meet a client's specific computing requirements; the software was hard-coded into the machine so, along with selling the equipment, IBM included the services of its engineers for a yearly fee. Minicomputers, on the other hand, were usually sold without engineering support, they were not customized and had to be programmed by whoever bought them. Consequently, the idea of a computer being used by a single person was unthinkable at that time. What ended up making the PC appealable for consumers was not the hardware that early computer hobbyists were so fond of tinkering with, but *software*—along with IBMs (cautiously skeptical) decision to finally enter the PC market.

PCs were from the outset all-in-one general purpose machines "ready to run" (Byte 1995, p. 100). By late 1977, the pioneering "trinity" (see Byte 1995; Williams and Welch 1985) of personal computers— the Apple II, the Tandy/RadioShack TRS-80, and the Commodore PET —had opened the market for a new class of cultural product: *software applications* for business, education, and entertainment. A whole new industry emerged around software—particularly around computer games—that would end up redefining human culture at large.[14] The consumer software industry would play a crucial role in the emergence of the UCD paradigm and UX.

In August 1981, IBM officially entered the PC market; this meant personal computing was finally legitimized by a "serious" (i.e., conservative) corporation willing to bet on the new technology. Whereas the "trinity" had certainly gained followers

[14] Arguably, the software industry continues to play a critical role in technological adoption. We should remember the success of smartphones depends mainly on the fact that they allow users to run myriads of "apps" controlling an equal amount of services.

in the electronics enthusiasts and educational markets before IBM introduced the Model 5150 PC most business users who had hesitated to buy an Apple or a Tandy (the Commodore was seen mainly as an educational device) were finally convinced. To the news media, unaware of the cultural origins of this technological shift, the computer was an overnight phenomenon whose success surprised even IBM itself (Campbell-Kelly et al. 2014, p. 248).

Engelbart's "electronic office" and Kay's Alto were two technological models that joined to form not only the modern GUI but also the paradigm of contemporary HCI (Campbell-Kelly et al. 2014, p. 259). Companies such as Apple and Microsoft capitalized on these innovations, "liberating" consumers from having to interact with the command line and creating a market for software applications which brought new challenges for the field of HCI and set the stage for the emergence of UCD and UX as disciplines. Before looking at the origins of these design paradigms, it is critical to focus on the ideas behind the GUI, in particular on its pedagogical imperatives, for it is there that we will find the reason why HCI researchers stopped treating the actual needs of end-users as an afterthought.

As we will see further along the way, one of the tenets of contemporary design practices following UCD approach (particularly Interaction Design) is creating technological solutions that are not only usable but *useful*. The goal is providing users with the means to accomplish something better; the technical solution is, therefore "just" an enabler, an affordance that will improve a user's experience while carrying out a task. To achieve this goal, designers need to understand the role of products in the context of meaningful activities, this means learning not only what kind of tasks a user engages in, but why she does it.

5.5 The Pedagogical Role of the GUI

Kay's goal was offering people, particularly children, not (just) a multipurpose tool, but a "metamedium" (Kay and Goldberg 1977) for constructing knowledge. Whereas other HCI pioneers such Engelbart and English (1968) had focused on improving HCI to "augment human intellect", Kay was looking instead to develop an *enabling* device for personalized *learning* (Coyne 1995, p. 33). Kay's vision highlighted the nature of the computer as prefigured by Alan Turing. Turing (1937) imagined his machine as capable of simulating, or rather of "computing"[15] any machine that was computable. Kay thought this universality—this capacity to simulate—could be extended to sound and images (Manovich 2013). Hence, he made simulation the "central notion" guiding the design of his prototypes, particularly, of the *Dynabook* (Kay and Goldberg 1977, p. 36).

If Kay and his team spent over a decade researching the computer's potential as "a medium for expression through drawing, painting, animating pictures, and compos-

[15] Turing never used the word "simulation" in his paper, the term is the product of later interpretation of Turing's ideas.

ing and generating music" (1977, p. 31), it was not due to artistic inclinations. Kay was interested in improving human learning potential through computational technology, but he disagreed with the prevailing rationalist conceptualizations of knowledge shared by most HCI researchers. For them, computers could be at best devices for capturing and retrieving information (Bush 1945) or machines for automating routine work (Licklider 1960). Whereas Kay regarded the computer as a "culture machine"—to borrow Manovich's formulation (2013); as a medium through which active learning and experimentation could be significantly amplified by simulation.

Influenced by the ideas of Jerome Bruner, Seymour Papert and Marvin Minsky, Kay, and his group at Xerox PARC imagined the computer interface as something that should be equally approachable for anyone regardless of age and prior cognitive skills and knowledge. In 1968, as a graduate student, Kay met the ideas of Papert through Minsky (Kay 2017b). Papert, who studied with developmental psychologist Jean Piaget, had realized that children under 12 years old are not well equipped to do "standard" symbolic mathematics, but they nonetheless could do other kinds of mathematical thinking when presented in a way that matched their current capacities (Kay 2002). Kay later came into contact with Jerome Bruner's interpretation of Piaget's ideas on children's cognitive development and came to believe that interaction with a computer interfaces should take advantage of the three "mentalities" (modes of representation) Bruner had identified: enactive (manipulate objects), iconic (recognize things), and symbolic (abstract reason); as opposed to merely stimulating the symbolic mentality like the traditional command line interface (CLI) did (Kay 2002; see also Manovich 2013, pp. 97–98). Kay condensed his vision in the slogan "doing with images makes symbols" (2002, p. 128), which culminated in the Alto's GUI.

Early programmers and HCI researchers were mostly mathematicians and scientists, their approach to interface design and programming, in general, was based on mathematical logic. A shift in paradigm required, to borrow Kay's formulation, "a new class of artisan" (1984, p. 54). These artisans understood the role aesthetics plays in cognitive processes, specifically one that privileged simplification via visual metaphors and analogies over abstract logical descriptions. To embrace this new paradigm required accepting that people are different from computers; that human behavior is far more complex than any logical model would admit. Therefore, a new design approach was required. One that was "pluralist" (interdisciplinary) enough to accommodate all the nuances of human behavior, and sensitive enough to place the human needs rather than the system's needs at the start and the core of the design process. This approach was UCD.

5.6 User-Centered Design (UCD)

The origins of UCD date back to the early 1980s, to the Project on Human-Machine Interaction from the Institute for Cognitive Science at the University of California, San Diego. At the time, a group of researchers from AI and psychology, among them Donald A. Norman, put together an interdisciplinary team of researchers and organized a series of conferences and workshops that culminated in the book *User Centered System Design* (1986). Both the name of the book[16] and the holistic approach it advocated grew in popularity among HCI practitioners and researchers and has since then become the dominating paradigm, particularly in Interaction Design (IxD). Another key document is "Designing for usability" by Gould and Lewis (1985), an article that outlined the main ideas and reasons for adopting an empirical approach in what was then called system design, that includes user research and intensive cycles of prototyping and testing.

The emergence of UCD is arguably a continuation of the ideas that led to the GUI in the first place, albeit more pragmatic and with the benefit of having computers already transformed into consumer products. Its origins may be attributed to HCI practitioners and researchers realizing that computers are not (just) about technology but about people using them. These researchers recognized that "computation is a social act"—to borrow Turkle's (1980, p. 22) words, and hence, that the computer could be understood as a social tool (Norman et al. 1986, p. 2) that influences social interactions and policy. They understood that the computer could and should be viewed "from the experience of the user", which is itself influenced by the nature of the task, the user herself, and the context of use.

The ideal driving the shift toward UCD was giving users "the feeling of 'direct engagement'" (Norman et al. 1986, p. 3). That is, the feeling that the computer itself receded to the background while letting the task at hand, whether it involved sound, words, or images to come to the forefront. This stance was radical insofar as it proposed completely subordinating the interface to "social concerns": to the various ways in which the computer could be use, rather than the other way around—as had been the case until then. So much so that Gould and Lewis (1985, p. 301) note that while they had been promoting these principles since the 1970s and many designers claimed not only that they applied them but that these ideas were "common sense", the reality is they did not even understand them. It has been more than 30 years since Gould and Lewis and Norman and Draper (1986) outlined the core principles of UCD, but only in the last decade or so have they been accepted and implemented in product design (Still and Crane 2017, p. 19).

5.7 What Is UCD

UCD developed from many different sources; it is related to Interaction Design (IxD) and User Interface Design (UID),[17] but whereas these are "artifact driven" notions, UCD is better understood as a comprehensive *process* (Wallach and Scholz

[16] The name "User Centered System Design" was originally an alliteration of the abbreviated name of the University of California, San Diego (UCSD). Norman and Draper (1986 iX) credit Paul Smolensky with having come up with the idea.

[17] UCD is informed by these two design areas but also by cognitive science, user research psychology, HCI, and ergonomics (Still and Crane 2017).

2012)—although not as ample as UX. UCD is a *cluster*[18] of operations comprising a framework that implicitly recognizes the interface of a computational device as a sociotechnical intersection. That is, as something where "many different kinds of things: people, machines, tasks, groups of people, groups of machines, and more" (Norman et al. 1986, p. 5) come together. As Wallach and Scholz (2012) note, there is little doubt that Gould and Lewis (1985) laid the foundational concepts and general approach on which current UCD practices are still based. This is no small feat, considering that in terms of technological development, three decades is a significant time span. Their central claim was that "[a]ny [computational] system designed for people to use should be easy to learn (and remember), useful, that is, contain functions people truly need in their work and be easy and pleasant to use" (1985, p. 300). They were, in short, advocating that to provide learnability, usability, and "delightful" experiences (see Cooper et al. 2014; see also Norman 2013), designers[19] ought to first and foremost *understand* their potential users.

Gould and Lewis do not define "usability",[20] however, their claim is echoed in the International Organisation for Standardisation (2018), according to which:

usability
[is the] extent to which a system, product or service can be used by specified users to achieve defined goals with effectiveness, efficiency, and satisfaction in a specified context of use.

Gould and Lewis (1985, p. 306) were prescient enough to recognize that in this age the product is not (just) the device but the *interface*. They understood the need to develop a robust methodology to increase usability, which undoubtedly would have a powerful impact on the emerging market of computational devices. They advocated three principles that are now obvious for anyone acquainted with UCD, but which at the time seemed if not foreign, at least superficial. First, that "systems designers" should engage in "interactive design" (1985, p. 302), that is, they should focus on the users and their tasks from the outset, understanding who they are and the nature of the work they engage in by studying their behavior through *direct contact*. Second, carry out empirical measurement while testing prototypes with actual users early in the design process, focusing on their reactions and suggestions. Third, embrace iterative cycles consisting of design, testing, and redesigning.

[18] A "cluster" is best understood as "a number of things growing naturally together" (Harper 2019).

[19] It is important to remember that, at the time, there was no such thing as a "user interface designer". The people designing interfaces were mostly self-taught programmers and scientists with little or no training in design. Design as practice continued to be primarily analogical and divided along two traditional branches: graphic and industrial.

[20] For thorough discussions on the history and broader meanings of the term see Johnson et al. (2007) and Sullivan (1989).

5.8 Understanding Users

Gould and Lewis further clarify that by understanding "typical users"[21] they do not mean identifying, describing, or stereotyping. They argue contact with them should be direct, preferably through interviews carried out *before* the actual design cycle begins because it is at that moment that the information gathered can influence the outcome of the design. This process stands in stark contrast to what inexperienced designers (mainly students) attempting to follow an empirical approach do: conducting user research *after* creating the prototypes thus falling into the trap of post hoc rationalization that forcibly attempts to validate design decisions that were already implemented. Gould and Lewis (1985, p. 302) note this type of user involvement resembles participatory design, a methodology that originated in Scandinavia and which advocates direct user involvement throughout the entire design process (for a thorough discussion see Luck 2018; see also Spinuzzi 2005).

Regarding "empirical measurements", Gould and Lewis are talking about testing and measuring variables such as learnability and usability with a user interacting with a prototype, rather than carrying out simple analytical questions. In other words, they warn against attempting to "sell" a finished interface to a potential user. Usability testing helps overcome the problem of designers being too used to their product and hence not being able to see all the potential pitfalls and untested assumptions in their project. Usability testing makes explicit the differences between the ways a designer and a user think about the interface.

Gould and Lewis understand iterative prototyping as an effective way to address the unpredictability of user's needs, which often lead to fundamental changes in design. Prototyping should be based on user testing, for the latter can reveal that even the most thoughtful design might prove to be inadequate. This implies that the implementation should be as flexible as possible, extending throughout the system. An essential aspect of iterative prototyping is that designers need to be capable of accepting (and reacting upon) test results that call for radical changes in the design and be prepared to "kill their darlings". In sum, testing prototypes can help designers to reliably identify critical problems in what they create; hence, it should not be treated as a luxury or unnecessary waste of time.

5.9 Norman's Approach

Gould's and Lewis' principles are aligned with one of Norman's most influential works, *The Design of Everyday Things* (2013). Norman argues his approach concerns three major areas of design: Interaction Design, Industrial Design, and Experience

[21] Gould and Lewis (1985, p. 302) suggest user research should be conducted for example with secretaries and clerks: from a historical standpoint this suggestion is interesting because it reveals an understanding of who the actual users of personal computers were. A knowledge that contradicts the naive assumptions that some computer manufacturers such as Xerox had at the time.

Design. However, whereas Gould and Lewis claim their approach could increase “usability marks” (and therefore make systems easier for users to learn and use), Norman’s approach is more holistic; he sees the users from a broader perspective. He talks about human–technology relations, not restricting his approach to a specific technology or context of use. Norman’s vital contribution is suggesting that designers not only provide a given product or service but a whole *experience*; something with an active aesthetic component. Experience, Norman (2013, p. 10) contends, is critical because it determines how people are going to internalize their interaction with a given technology.

For Norman, the design is ultimately a humanistic activity; a form of mediation. As he puts it, “[a]ll artificial things are designed” (2013, p. 4); the design is concerned with how things work and how they are controlled, and thus how humans interact with them. But while people build machines, their behavior is limited (procedural and literal) and may often seem alien to the users. Traditionally, it was users who had to adapt to the situations presented by the machine, but that should not be the case any longer. Instead, machines should adapt to people’s needs and circumstances, and it is the designer’s task to make sure that happens. The problem, Norman contends, is that the people in charge of designing the technologies are experts in the machines, not in people’s behavior. Furthermore, they are often convinced that logic is the most appropriate way of thinking, whereas human thought is far more complex. The technological design thus stands at an intersection between humans and machines; its task is bridging the gap that separates the two. For Norman (2013, p. 9) UCD, or rather HCD is not only an approach but a design *philosophy* that relies primarily on (scientific-like) observation of people. Because specifying what is going to be defined is the most challenging aspect of the process UCD/HCD instead iterates potential approximations to the solution. UCD is a methodology that can be employed by different design areas, regardless of their specific focus (e.g., Interaction, Communication, Industrial products).

In the decade since Norman and Gould and Lewis first promoted their ideas UCD/HCD has been expanded and adapted by many design practitioners, leading to somewhat different methodologies which, nonetheless, maintain the same basic principles: understand the user before designing and incorporate insights from user research throughout the design process; test prototypes in recursive iterative cycles and be prepared to make changes after each cycle, regardless how radical they need to be.

In summary, UCD is a process or rather a set of processes that emphasize an approach to design that breaks from the traditional product-centric/technological approach by taking care of the whole experience of the user. It is a humbling method that highlights the uncertain nature of design, and the complexities of its various stages, putting a humane focus throughout the design process. UCD is above all iterative; it implicitly addresses a vital issue for design practitioners, which Parsons (2015) calls the “epistemological problem” or difficulty of design: the question of how a designer can know her solution is going to solve the intended problem (see also Galle 2011). UCD tackles this problem by adopting a fundamentally empirical method to gain as much information from the users as possible to craft a unique experience. How this concept should be understood in the context of design and what is its relationship with aesthetics will be the focus of the next section.

5.10 User Experience (UX)

Contemporary design practices address complex problems that involve difficult sociocultural issues and reveal the deep entanglement between human behavior and technologies. If we lend credit to Norman, designers today are more like applied behavioral scientists than applied artists. New design areas such as interaction or product design require a deep "understanding of human cognition and emotion, sensory and motor systems, and sufficient knowledge of the scientific method, statistics, and experimental design" (Norman 2010). Traditional design skills such as drawing, sketching, and modeling are supplemented and sometimes replaced by programming, and scientific methodologies for gathering and analyzing data. Design products and what is expected from them have thus become significantly more complex.

The emergence of UX is arguably the result of the technological shift discussed in the first section of this chapter, which led designers from "merely" designing concrete objects (i.e., "stuff") to designing the *conditions* that may elicit a positive and complex response from users. In the early and mid-decades of the past century, designers mainly focused on "external" aspects of products, i.e., their form, function, use, and materials (Buchanan 2001). However, with the arrival of the PC and consumer software, designers began to move their focus away from "visual symbols and things" and onto understanding products "from the inside" of the humans interacting with them in specific social and cultural circumstances. Computational technology opened an uncharted space for a design where form, function, use, and materials are still important, but they are re-conceptualized through research attempting to understand what is it that makes a product useful, usable, desirable and delightful (2001, p. 13).

Defining UX in general terms is a difficult if not impossible task. It can be a practice or area of focus in contemporary design but also the *result* of a design process.[22] It is an umbrella term that attempts to describe all the complex things that a user undergoes when interacting with a designed artifact. Hence, while it is a relatively novel concept, it describes phenomena that have been discussed for a long time by designers under other names, such as ergonomics, affordances, or anthropometrics. The main distinction, however, is that unlike previous notions, UX explicitly acknowledges that whatever happens between a human being and a designed artifact has a strong aesthetic component.

[22] From this point onwards, UX will be used to designate the result of a design process, whereas UXD will refer to the activity aiming to achieve it.

To the best of our knowledge, Norman was among the first one to use—if not the inventor[23] of—the term "User Experience" (Norman 2013, xiii–xiv) in the early 1990s while he was the head of the "User Experience Architect's Office" at Apple. Norman implicitly defines experience as "the aesthetics of form and the quality of interaction" provided by a given product (Norman 2013, p. 4). This implies the product is not only usable but useful; that its features are immediately discoverable and understandable to the user. This succinct definition is a good starting point. Nonetheless, to fully grasp what experience stands for in a contemporary design, we need to look at its origins and evolution as a concept and its relationship with aesthetics. But also, to its usage within a philosophical school (American pragmatism) and, particularly, in the work of John Dewey. This we will do before turning our attention to the ways experience influenced computer system design, HCI, UCD, and IxD.

From a (traditional) epistemological standpoint, an experience is that which contrasts to what is thought or to what is accepted based on authority or tradition; it is what we perceive through our senses; information that comes from external sources (or through inner reflection) (Bunnin and Yu 2009, p. 240). In this sense, the experience is associated with empirical observation. Because it concerns sensory perception experience is closely linked to aesthetics, which was initially understood as "the science of sensitive knowing" (Bunnin and Yu 2009, p. 17), from the Greek *aisthitiki*, "perceived by the senses" (Fishwick 2006).

Although aesthetics is usually associated with art, there is an essential distinction between the two. Aesthetics may be concerned with works of art, but it is not restricted to art or beauty or the beautiful (Nake 2012, p. 66). Aesthetics is also concerned with value and with our experience of the environment (both natural and artificial); it is an autonomous branch of philosophy concerned with the analysis of problems relating to perception. It was initially conceived as a companion and complement to logic, and thus its focus was human cognition. Whereas logic studied discursive and rational cognition, aesthetics focused on holistic sensory cognition (*cognitio sensitiva*), that is, cognition experienced and practiced through our senses, tied to our physical capacities (Proudfoot and Lacey 2010). There are many approaches to aesthetics, but the one that interests us, due to its lasting influence on contemporary design practices and areas of specialization such as interaction design and User Experience Design (UXD) is by the American Pragmatist philosopher, John Dewey.

5.11 Dewey and Pragmatism

Pragmatism is a philosophical school that emerged in the United States in the late nineteenth century. Unlike other philosophical strains in the Western Tradition, pragmatism evaluates claims (e.g., concerning meaning, truth, knowledge, or morality)

[23] Norman admitted to having "invented" the term because usability and human interface seemed too narrow to account for everything that happened when a person interacted with a system (Merholz 2007).

not in terms of perennial axioms or syllogisms but in terms of the consequences that a given action has (Dusek 2009). Pragmatism rejects dualism (mind vs. body distinction) and the separation of theory and practice; it embraces the materiality of the world, the embodiment of knowledge, the interaction of the senses, and the formative power of technology in everyday life (Coyne 1995, p. 17). Pragmatism is anti-essentialist; it emphasizes practice, not representation (Ihde 2009). For pragmatism, experiences are crucial for creating knowledge. Their view of experience is holistic and dynamic, according to it, humans do not merely (passively) receive individual sense impressions but actively engage with the world through active habits; hence we continuously transform our experience of it (see Pihlström 2011, p. 31).

According to Coyne (1995, pp. 38–41) Dewey understood facts, ideas, and concepts *as tools*; he regarded theoreticians as technicians. Tools are not universally useful; their applicability changes according to the situation. Thus, he did not grant any special privilege to reason or inference—he regarded science as just another form of practice, albeit highly specialized. Knowledge cannot exist outside of doing, knowing is "knowing *how"* instead of "knowing that". Humankind, for Dewey, is not above nature but always involved *with* nature and in constant interaction with it; life happens not only in an environment but within it. More important, and because he emphasizes (human action) Dewey regarded perception not as analytical or passive but as a participatory activity, and this is key for his understanding of aesthetics.

For Dewey aesthetic artifacts such as works of art have no intrinsic, essential features, the "art" is in what the object does *within an experience*. To understand the aesthetic value of an artifact, we need to look at ordinary, "in the raw", everyday aesthetics. That is to say, for example, that if we want to understand the Parthenon, we first must understand the cultural context of Athenian society (Leddy 2016).

In Dewey's view, aesthetic experiences begin in happy absorption in an activity (poking fire in a campfire, or watching a baseball game), so a crafty mechanic fixing a car may be in this sense "artistically engaged" (see Granger 2006). Organisms (including humans) engage in a dialectical relationship with their environment: every creature has needs, their life flows are a constant rhythmic resolution of tensions between requirements and their satisfaction; between disunity and a unity (balance) (Leddy 2016). For humans, this rhythm is conscious. Direct experience is a function of the interaction between us and our environment. The aesthetic experience involves a drama (narrative) where actions, feelings, and meaning play a part. The most intense aesthetic experiences happen in the transitions between the disturbance of needs and the harmony of balance when needs are met. Happiness is the result of deep fulfillment; when every aspect of our being is adjusted to the environment (in full balance). Experience is the result of active engagement with these tensions when we infuse them with conscious meaning through communication (Leddy 2016). Consequently, the experience is not only the result of the interaction between subject and environment but the subject's reward when it transforms mere interaction into active (meaningful) participation.

Dewey's book (1980) *Art as Experience* and, in particular, the chapter "Having an experience" has had a longstanding influence on contemporary design practices, particularly on IxD (Buchanan 2009). Dewey's book was compulsory reading in the Industrial Design course at the New Bauhaus in Chicago (Findeli 1990). It later informed HCI research at Xerox PARC and continues to elicit much discussion among IxD researchers (Dixon 2019).

For Dewey, in "an experience" the material of experience is fulfilled or consummated, e.g., when a game is played, or a problem is solved (Leddy 2016). As we saw before, Dewey understands life as a collection of histories, each one with a unique quality. In an experience, every one of its components follows in an unbroken chain without sacrificing their identity; each part is a phase of an enduring whole. A good example of experience is an artwork; in it, separate elements participate in forming a unity; their particular identity is not diluted but enhanced.[24] For Dewey, no experience has unity without aesthetic quality, although this does not imply that *all* experiences can be reduced to aesthetic experiences (Leddy 2016). Emotion is the unifying quality that distinguishes an aesthetic experience from other kinds of experiences (Buchanan 2009).

5.12 Experience as Design

Dewey's influence is palpable in the work of Hassenzahl (2010, pp. 5–30), who describes experiences in terms of unique *emergent* qualities that are not reducible to (nor explainable by) their constituting elements and processes. Nonetheless, these elements are open to study and deliberate manipulation; experiences can thus be shaped through carefully modifying their elements. Experiences are lived episodes comprising sights, sounds, feelings, thoughts, and actions; they are stories emerging from the "dialogue" of a person with her surroundings. Experiences are holistic, situated, and dynamic; they arise from the activation of perception, action, motivation, and cognition at a given place and moment, and they extend over a certain timespan. Experiences may occur in infinite variations, but, in Hassenzahl's view, there are universal psychological needs that are essential constituents of experience (2010, p. 57).

UX is a sub-category of experience that is deliberately elicited and shaped through an (interactive) product (Hassenzahl 2013). UX is not unlike experience at large; the difference being that it focuses on a person's attention on that specific product. The product is not the experience per se, but a facilitator, a mediator that can shape or influence what and how we experience a given activity (2010, p. 8). The emergence of a given experience cannot be guaranteed; however, careful application of knowledge about how experiences are elicited can make them more likely, that is precisely the task of UXD.

Although for UX, the interactive product is a necessity, UXD is *not* about the technology itself, but about transcending its materials, about making it an instrumental

[24] Artworks may be understood as integral complex informational systems, wherein each element is necessary for the aesthetic content to emerge, a slight change in the configuration may alter the whole meaning of the piece (see Hernández-Ramírez 2016).

and yet almost transparent presence. The technologies are the canvas, the pretext for the UX designer. Given that the majority of these products are digital, an excellent way to understand UXD is in terms of narratives, or "material tales" narrated through digital objects (Hassenzahl 2013). Because experiences are dynamic and happen over a timespan, any given moment within that timespan can impact the overall experience. Designers can influence that experience by manipulating order and timing; by *scripting* interactions among the elements (Hassenzahl 2010, pp. 29–30).

Products fulfill needs, but to do so, they need to be "instrumental", i.e., able to shape the user's experience as intended (Hassenzahl 2013). Products need to be functional, useful, discoverable, and understandable to satisfy a particular need; it is only then that a (good) experience emerges. Functionality and usability, however, need to be contextualized, and that means being meaningful. A genuinely unique experience requires that not only the engineering, manufacturing, and ergonomic aspects are met, but also the aesthetic ones; interaction with the product should also be delightful and enjoyable. It is only with this holistic satisfaction of needs that truly unique experience can emerge (Hassenzahl 2010; Norman 2013).

5.13 Concluding Remarks

Pervasive computation and general advances in hardware and software have allowed us to transform artifacts that were traditionally "dead" into alive devices. Computational objects have come a long way since the dawn of HCI, UCD, and UX. While ergonomics and HCI emerged almost at the same time as computational technology, only with the democratization of computers, it became a necessity for designers to honestly think about their users. Smart technologies and what we end up defining and recognizing as robots will determine the practical principles, and the type of experience designers will be able to shape.

Robots should be able to enhance and improve our experience of the world, improve our living standards, liberate us of chores and burdens so we can dedicate ourselves to cultivate meaningful activities. Robots and our technologies, in general, are reflections of what we are, how we understand them and design them reflect our understanding of ourselves. Designing should always consider the human–technology relationship as complementary, not in terms of substitution. We need artificial agents that highlight what is valuable and enjoyable about being human. Ultimately, and given the broader objectives of the volume to which this chapter belongs, it is fundamental to pay attention to the core principles behind UCD and UX. It is crucial to listen to the core ideal of UCD and UX: to focus on the human in her context, not in decontextualized technology for the sake of technology.

References

Alesso HP, Smith CF (2008) Connections: patterns of discovery. Wiley, Inc., New Jersey. https://doi.org/10.1002/9780470191538

Archer B (1979) Design as a discipline. Des Stud 1(1):17–20. https://doi.org/10.1016/0142-694x(79)90023-1

Brey P (2008) The computer as cognitive artifact and simulator of worlds. In: Briggle A, Waelbers K, Brey PA (eds) Current issues in computing and philosophy, vol 175. IOS Press, Amsterdam, pp 91–102

Brown T (2008) Design thinking. Harv Bus Rev 85–92. https://hbr.org/2008/06/design-thinking

Brown T (2009) Change by design: how design thinking transforms organizations and inspires innovation. Harper Collins e-books, New York; London

Bryson JJ (2009) Robots should be slaves, May 21. http://www.cs.bath.ac.uk/~jjb/ftp/Bryson-Slaves-Book09.html. Accessed 22 Dec 2018

Bryson JJ (2019). The past decade and future of AI's impact on society. In: OpenMind. BBVA, Madrid, Feb 27. https://www.bbvaopenmind.com/en/articles/the-past-decade-and-future-of-ais-impact-on-society/. Accessed 14 May 2019

Buchanan R (2001) Design research and the new learning. Des Issues 17(4):3–23. https://doi.org/10.1162/07479360152681056

Buchanan R (2009) Thinking about design: an historical perspective. In: Meijers A (ed) Philosophy of technology and engineering sciences, vol 9. Elsevier, North Holland, Amsterdam; London; Boston, pp 409–453

Buley L (2013) The user experience team of one: a research and design survival guide (Revised and Expanded.). Rosenfeld Media. https://books.google.pt/books?id=aHo3DwAAQBAJ

Bunnin N, Yu J (2009) The Blackwell dictionary of Western philosophy. Wiley-Blackwell, Oxford

Bush V (1945) As we may think. The Atlantic. http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/

Byte (1995) Most important companies. Byte 20(9, Special Issue):99–103. https://archive.org/details/byte-magazine-1995-09

Campbell-Kelly M, Aspray W, Ensmenger N, Yost JR (2014) Computer. A history of the information machine, 3rd edn. Westview Press, Boulder, Colorado

Carroll JM (2013) Human computer interaction—brief intro. In: Dam RF, Soegaard M (eds) The encyclopedia of human–computer interaction, 2nd edn. The Interaction Design Foundation, Aarhus, Denmark. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro

Ceruzzi PE (2003) A history of modern computing, 2nd edn. The MIT Press, Cambridge, Massachusetts

Cooper A, Reimann R, Cronin D, Noessel C (2014) About face: the essentials of interaction design, 4th edn. Wiley, Hoboken

Coyne R (1995) Designing information technology in the postmodern age: from method to metaphor. The MIT Press, Cambridge, Massachusetts

Cross N (2011) Design thinking: understanding how designers think and work. Berg, Oxford; New York

Danaher J (2018) Toward an ethics of AI assistants: an initial framework. Philos Technol 31(4):629–653. https://doi.org/10.1007/s13347-018-0317-3

Dewey J (1980) Art as experience, 23rd edn. Perigee Books, New York

Dixon B (2019) Experiments in experience: towards an alignment of research through design and John Dewey's pragmatism. Des Issues 35(2):5–16. https://doi.org/10.1162/desi_a_00531

Dusek V (2009) Introduction: philosophy and technology. In: Olsen JKB, Pedersen SA, Hendricks VF (eds) A companion to the philosophy of technology. Wiley-Blackwell, Chichester, UK; Malden, Massachusetts, pp 131–140. https://doi.org/10.1002/9781444310795.ch22

Engelbart DC (1962) Augmenting human intellect: a conceptual framework (AFOSR-3223 Summary Report No. SRI 3578). Stanford Research Institute, Menlo Park, California. http://dougengelbart.org/content/view/138. Accessed 24 May 2019

Engelbart DC, English WK (1968) A research center for augmenting human intellect. In: Proceedings of the '68 fall joint computer conference. American Federation of Information Processing Societies (AFIPS); ACM Press, San Francisco, California, pp 395–410. https://doi.org/10.1145/1476589.1476645

Findeli A (1990) Moholy-Nagy's design pedagogy in Chicago (1937–46). Des Issues 7(1):4–19. https://doi.org/10.2307/1511466

Fishwick P (2006) An introduction to aesthetic computing. In: Fishwick P (ed) Aesthetic computing. The MIT Press, pp 3–27

Floridi L (2002) On the intrinsic value of information objects and the infosphere. Ethics Inf Technol 4:287–304. https://doi.org/10.1023/a:1021342422699

Floridi, L. (2013). Present and foreseeable future of artificial intelligence. In: Sabba F (ed) Estratto da noetica vs informatica: Le nuove strutture della comunicazione scientifica. Atti del convegno internazionale, roma, tempio di adriano. Firenze, pp 131–136

Floridi L (2014) The fourth revolution: how the infosphere is reshaping human reality. Oxford University Press, Oxford, UK

Galle P (2011) Foundational and instrumental design theory. Des Issues 27(4):81–94. https://doi.org/10.1162/desi_a_00107

Garvey C (2018) Broken promises and empty threats: the evolution of AI in the USA, 1956–1996. Technology's Stories. https://doi.org/10.15763/jou.ts.2018.03.16.02

Gould JD, Lewis C (1985) Designing for usability: Key principles and what designers think. Commun ACM 28(3):300–311. https://doi.org/10.1145/3166.3170

Granger DA (2006) John Dewey, Robert Pirsig, and the art of living: revisioning aesthetic education. Palgrave Macmillan US, Basingstoke, Hampshire; New York. https://doi.org/10.1007/978-1-137-12252-0

Harper D (ed) (2019) Definition of "cluster". Online Etymology Dictionary. https://www.etymonline.com/word/cluster#etymonline_v_13860. Accessed 31 May 2019

Hassenzahl M (2010) Experience design: technology for all the right reasons. (S. L. on Human-centered informatics, Trans.), vol 3. Morgan & Claypool Publishers LLC, pp 1–95. https://doi.org/10.2200/s00261ed1v01y201003hci008

Hassenzahl M (2013) User experience and experience design. In: Dam RF, Soegaard M (eds) The encyclopedia of human–computer interaction, 2nd edn. The Interaction Design Foundation, Aarhus, Denmark. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/user-experience-and-experience-design

Hernández-Ramírez R (2016) Overcoming information aesthetics: in defense of a non-quantitative informational understanding of artworks. J Sci Technol Arts 8(1):17–27. https://doi.org/10.7559/citarj.v8i1.218

Hernández-Ramírez R (2018) On design thinking, bullshit, and innovation. J Sci Technol Arts 10(3):2–45–57. https://doi.org/10.7559/citarj.v10i3.555

Ihde D (2009) In: Langsdorf L (ed) Postphenomenology and technoscience: the peking university lectures. SUNY Press, Albany, New York

ISO EN 9241-11 (2018) Ergonomics of human-system interaction—part 11: usability: definitions and concepts. International Organization for Standardization. https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en. Accessed 3 May 2019

Johnson DG, Verdicchio M (2017) Reframing AI discourse. Minds Mach 27(4):575–590. https://doi.org/10.1007/s11023-017-9417-6

Johnson J, Roberts TL, Verplank W, Smith DC, Irby CH, Beard M, Mackey K (1989) The xerox star: a retrospective. Computer 22(9):11–26. https://doi.org/10.1109/2.35211

Johnson RR, Salvo MJ, Zoetewey MW (2007) User-centered technology in participatory culture: two decades 'beyond a narrow conception of usability testing'. IEEE Trans Prof Commun 50(4):320–332. https://doi.org/10.1109/tpc.2007.908730

Kaptelinin V (2013) Activity theory. In: The encyclopedia of human–computer interaction, 2nd edn. The Interaction Design Foundation, Aarhus, Denmark. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/activity-theory. Accessed 2 Nov 2018

Kaptelinin V, Nardi B (2012) In: Carroll JM (ed) Activity theory in HCI: fundamentals and reflections, vol 13. Morgan & Claypool Publishers. https://doi.org/10.2200/s00413ed1v01y201203hci013

Karjaluoto E (2013) The design method: a philosophy and process for functional visual communication. New Riders, New York

Kay AC (1969). The reactive engine. PhD thesis, The University of Utah, The University of Utah. http://www.chilton-computing.org.uk/inf/pdfs/kay.htm

Kay AC (1984) Computer software. Sci Am 251(3):52–59. https://doi.org/10.1038/scientificamerican0984-52

Kay AC (2002). User interface: a personal view. In: Packer R, Jordan K (eds) Multimedia: from wagner to virtual reality. Norton, New York, pp 121–131. https://books.google.pt/books?id=LKEECCdHLqQC

Kay A (2017a) What was it like to be at Xerox PARC when Steve Jobs visited? Quora, June 13. https://www.quora.com/What-was-it-like-to-be-at-Xerox-PARC-when-Steve-Jobs-visited. Accessed 13 June 2017

Kay AC (2017b) Remembering Seymour Papert. YouTube. Video, Jan 27. https://www.youtube.com/watch?v=XTbAdTRFp1k. Accessed 26 May 2019

Kay AC, Goldberg A (1977) Personal dynamic media. Computer 10(3):31–41. https://doi.org/10.1109/c-m.1977.217672

Leddy T (2016) Dewey's aesthetics. In: Zalta EN (ed) The stanford encyclopedia of philosophy (Winter 2016). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/win2016/entries/dewey-aesthetics/

Lee K-F (2018) AI superpowers. Houghton Mifflin Harcourt, New York. Accessed 17 Jan 2019

Licklider JCR (1960) Man-computer symbiosis. IRE Trans Hum Factors Electron HFE-1(1):4–11. https://doi.org/10.1109/thfe2.1960.4503259

Luck R (2018) What is it that makes participation in design participatory design? Des Stud 59:1–8. https://doi.org/10.1016/j.destud.2018.10.002

Manovich L (2002) The language of new media. The MIT Press, Cambridge, Massachusetts

Manovich L (2013) In: Ricardo FJ (ed) Software takes command. Bloomsbury Academic, New York

McCarthy J, Wright P (2004) Technology as experience. The MIT Press, London

Meinel C, Leifer L, Plattner H (eds) (2011) Design thinking: understand—improve—apply. Springer, Berlin, Heidelberg; New York; Dordrecht; London. https://doi.org/10.1007/978-3-642-13757-0

Merholz P (2007) Peter in conversation with Don Norman about UX & innovation. Adaptive Path, Dec 13. https://adaptivepath.org/ideas/e000862/. Accessed 24 Apr 2019

Nake F (2012) Information aesthetics: an heroic experiment. J Math Arts 6(2–3):65–75. https://doi.org/10.1080/17513472.2012.679458

Norman DA (2010) Why design education must change. Core 77, Nov 26. http://www.core77.com/posts/17993/why-design-education-must-change-17993. Accessed 22 May 2019

Norman DA (2013) The design of everyday things (Revised and expanded edition.). Basic Books, New York

Norman DA, Draper SW (eds) (1986) User centered system design: new perspectives on human—computer interaction. Lawrence Erlbaum Associates, New Jersey; London

Norman DA, Draper SW, Lewis C (1986) Introduction. In: Norman DA, Draper SW (eds) User centered system design: new perspectives on human–computer interaction. Lawrence Erlbaum Associates, New Jersey; London, pp 1–6

Norman DA, Miller J, Henderson A (1995) What you see, some of what's in the future, and how we go about doing it. In: Conference companion on Human factors in computing systems—CHI '95. https://doi.org/10.1145/223355.223477

Parsons G (2015) The philosophy of design. Polity Press, Cambridge; Malden

Pihlström S (ed) (2011) The continuum companion to pragmatism. Continuum, New York

Proudfoot M, Lacey A (2010) The routledge dictionary of philosophy, 4th edn. Routledge, New York

Saffer D (2010) Designing for interaction: creating innovative applications and devices, 2nd edn. New Riders, San Francisco

Smith DK, Alexander RC (1988) Fumbling the future: how xerox invented, then ignored, the first personal computer. William Morrow and Company, New York

Spinuzzi C (2005) The methodology of participatory design. Tech Commun 52(2):163–174

Still B, Crane K (2017) Fundamentals of user-centered design: a practical approach. CRC Press, Taylor & Francis Group, Boca Raton; London; New York

Sullivan P (1989) Beyond a narrow conception of usability testing. IEEE Trans Prof Commun 32(4):256–264. https://doi.org/10.1109/47.44537

Sutherland IE (1964) Sketchpad a man–machine graphical communication system. Simulation 2(5):R–3–R–20. In: Reprinted from the proceedings of the spring joint computer conference 1963, pp 329–346. https://doi.org/10.1177/003754976400200514

Turing AM (1937) On computable numbers, with an application to the entscheidungsproblem. Proc Lond Math Soc s2 42(1):230–265. https://doi.org/10.1112/plms/s2-42.1.230

Turkle S (1980) Computer as Rorschach. Society 17(2):15–24. https://doi.org/10.1007/bf02700055

Wallach D, Scholz SC (2012). User-centered design: why and how to put users first in software development. In: Maedche A, Botzenhardt A, Neer L (eds) Software for people: fundamentals, trends and best practices. Springer, Berlin, Heidelberg, pp 11–38. https://doi.org/10.1007/978-3-642-31371-4_2

Williams G, Welch M (1985) A microcomputing timeline. Byte 10(9, 10th Anniversary Issue), 198–208

Wright P, McCarthy J (2010) Experience-centered design: designers, users, and communities in dialogue, vol 3. Morgan & Claypool Publishers LLC. https://doi.org/10.2200/s00229ed1v01y201003hci009

Rodrigo Hernández-Ramírez studied communication, design & visual communication, and philosophy of science in his native Mexico City, where he worked as a designer and front-end developer for several years. He holds an M.A. in Multimedia Art with a specialization in photography from the University of Lisbon, where he also obtained a Ph.D. in Fine Arts with specialization in multimedia art. He is a member of the Centre for Research and Studies in Fine Arts (CIEBA) at the same institution. Currently, he teaches interaction design and design research at IADE, in Lisbon where he is an integrated member of UNIDCOM. His research interests stand at the intersection of philosophy of technology, design, and new media art. He is particularly interested in human–technology relations and how technologies shape the way we understand the world and ourselves.