xCoAx 2020
Computation,
Communication,
Aesthetics & X
2020.xCoAx.org
Graz, Austria

A Sketch of Some Principles for Good Design in The Age Of Smart Automation

Rodrigo Hernández-Ramírez
rodrigo.ramirez@universidadeeuropeia.pt
IADE, Universidade Europeia,
Lisbon, Portugal
UNIDCOM/IADE, Lisbon, Portugal

Keywords: AI Ethics, Complexity, Design Methodology, Design Principles, Smart Automation, Sociotechnical Systems, User-Centered Design, Via Negativa.

Over the last decades, technological advancements have allowed automation to become "smart" and thus capable of replacing human manual control, planning, and problem-solving in a growing number of activities. This cognitive outsourcing has improved people's lives in several ways, but it has also brought a host of new problems such as loss of privacy and human liberties, deskilling, new forms of exploitation, harassment, and increased inequalities. This paper begins with the assumption that these issues are the consequence of poor design, and therefore asks what good design in the age of smart automation is. It analyses the general inherent complexities of automation and the role that User-Centered Design, arguably the contemporary dominant paradigm in design practices, could play in taming these complexities. It does so to provide a rough sketch of principles that a humanist and ethically-minded design approach should follow to ensure our technologies meet the moral, political, and social needs of people in the present and near future.

1. Introduction

While automation is relatively old, over the past decades, it has become "smart" thanks to the combination of recent developments in AI methods, computing power, and data availability. Smart automation has turned vital for many aspects of human life. In the so-called developed regions of the world, virtually every area of human activity, from fabrication to healthcare, now integrate some form of smart, automated system. Widespread implementation of smart automation, however, has brought many problems, including loss of privacy, threats to democracy, new forms of discrimination and exploitation, as well as new avenues for abuse and harassment. While admittedly no technology is neutral or intrinsically beneficial for human life, the above problems are not necessarily a direct consequence of smart automation *per se*, but rather, of poorly designed implementations of it.

There are many ways to explain poor design, including lack of experience, ignorance, incompetence, lack of resources, or even straightforward malice. However, in the case of contemporary smart, automated systems, arguably one of the main culprits (along with the logic imposed by venture capital) is the mixture of overconfidence, monocultural biases, and technocentrism afflicting their designers. The people fitting this profile tend to focus mostly on the positive aspects of automation and rarely on their downsides; hence they usually fail to consider that any new automated product will be embedded within larger and increasingly complex socio-technical systems. This type of designers tends to care more about the success of their creations and the profit they will bring to them and other stakeholders than for the well-being of end-users. That is to say, the people that will suffer the consequences misuses, and unforeseen risks brought by the new technologies.

More important, people responsible for poorly designed smart automated systems fail to consider whether their creations should exist in the first place. How to counter these situations is a matter of grave concern nowadays, for it is in this day and age that we will set the pace and tone of the technological developments that will shape human life for the next decades. Some form of regulation might be useful to address these issues; however, the scale at which it should be implemented, and the nature of the standards adopted are still a matter of great debate. Not to mention that policy tends to be impractical on account of the fact that it usually lags behind technological innovation. That is why one of the central questions that people involved in the design of smart automation must address is how to create systems that genuinely improve people's lives while minimising potential negative impacts. In other words, they ought to think about what "good design" should mean in the age of smart automation.

This paper takes that question as a starting point. It begins with an overview of automation and the inherent problems that every automated system can have, regardless of whether it is smart or not. This is followed by a brief outline of User-Centred Design (UCD) principles and their role when dealing with complexity. Through an admittedly simplified sketch of principles for good design, this paper shows that the UCD approach should integrate a robust ethical stance. It argues designers should steer as much as possible away from blue-sky technological thinking and instead pursue a *via negativa* approach when designing. That is, designers should focus less on the positive aspects of their creations and think more about everything that could go wrong when they go out into the world.

This paper shows that designers should strive to make explicit their assumptions and biases about technological systems and the long-term impact they can have on users by focusing not on the technical and methodological obstacles impeding their design but on the ethical constraints they should impose. This paper argues that to become genuinely humanistic, UCD should enable users to achieve a better understanding of their systems, for a thorough knowledge of our tools goes a long way in helping us navigate the muddle that is life. Ultimately, this paper argues that design has a crucial responsibility in developing our human project for the future. It is important to note, however, that the principles here outlined are nothing but rough sketches and therefore still require considerable reworking before reaching the point of turning prescriptive. The exercise here developed should be seen as a starting point for developing a more thorough critique of contemporary design approaches in the face of smart automation.

2. The Paradoxes of Automation

Humans have been automating tasks for a long time. Arguably, in light of recent discoveries about other animal species' capacity to communicate, develop cultures, and use tools, a case can be made that our propensity to "off-load" work to artificial systems[1] is a crucial aspect of being human (Martinho-Truswell 2018). Traditionally, automation has involved delegating the execution of physical or non-cognitive tasks originally carried out by humans or animals (e.g., beasts of burden) to machines (Danaher 2018; Johnson and Verdicchio 2017). Examples of the former include a bow and arrow or a windmill. In the last decades, however, the development of computational tools in general, and Artificial Intelligence (AI) in particular, has enabled automated systems to take over cognitive and mental tasks. The aim of automated systems is thus no longer confined to carrying out physical work but also "to replace human manual control, planning, and problem-solving" (Bainbridge 1983, 775). Hence, broadly understood, smart automation may be seen as "cognitive outsourcing" (Danaher 2018) to artificial (human-designed) systems.

In principle, automated systems should be capable of accomplishing processes without human intervention during runtime—that is, after all, their reason to exist. Nonetheless, every form of automation requires a

1. A system may be understood as an entity that can be separated into parts, which are all simultaneously linked to each other in a specific way (Vermaas et al. 2011, ch. 5).

minimum of human supervision, if not to trigger their operation, at least to guarantee their functioning, to make adjustments, to provide maintenance, and to expand and improve their features (Bainbridge 1983). In the case of simple automated systems, the former tasks tend not to be a problem; but as systems grow in complexity, so does the oversight they require. Since it is human beings who make sure automated systems operate safely and are also responsible for mitigating the consequences of those systems' malfunctions, it is clear that the more complex the automation, the more crucial the role of the human becomes (Strauch 2018). In this sense, then, and regardless of how sophisticated they might be, automated systems paradoxically continue to be human-machine systems (Bainbridge 1983).

The rationale for adopting automated systems is that they are faster, more reliable, and efficient than human beings carrying out the same tasks; this tends to be true for the most part, particularly when it comes to repetitive actions. Ironically, however, errors introduced when automated systems are conceived can be a significant source of operating constraints and difficulties, and these operating problems usually end up being solved by human operators (Bainbridge 1983; Strauch 2018). Automation implies abstracting processes, breaking them down and formalising them into definite steps, and then organising them into specific sequences. In other words, automation implies a deep understanding of those processes; otherwise, it would be impossible to "tell a machine" how to simulate them. Tasks that follow explicit procedural logic—e.g., simple mathematical calculations—are easier to formalise that is why machines vastly exceed human speed, quality, efficiency, and accuracy when it comes to computing (Autor 2014).

Nonetheless, such is not the case with activities that require flexibility, fine motor skills, judgement, and common sense. These are tasks that humans accomplish with little effort (e.g., giving advice, preparing an omelette or coming up with a good joke), not because we know their explicit "rules", but because we have *tacit* knowledge of how to achieve them (Autor 2014). A direct consequence of this paradox is that when those responsible for conceiving automated systems intend to replace human operators, ironically, these operators end up carrying out the tasks that could not be automated (Bainbridge 1983). For example, people still need to load paper into copying machines, moderate content in social networks, and make all sorts of input readable for machines. To paraphrase anthropologist David Graeber (2018), the "problem" of automating tasks that are easy for humans to solve but difficult to automate is that doing so requires vast amounts of (semantic) labour carried out by humans to render them into a form that can even be recognised by a computer. Anyone who has ever tried to input data into an interface that lacks some form of natural language recognition might understand the frustration caused by having to "translate" everything for the computer.

2. A socio-technical system may be regarded as a hybrid system (i.e., one whose components belong to many different "worlds") with an extremely high degree of complexity that has many users at any given moment, and that involves people both as users and operators (Vermaas et al. 2011).

3. The problem, however, is that ML models are often inscrutable, meaning that those who conceived and trained them do not have a clear understanding of exactly how they operate.

Automation has been historically more prevalent in socio-technical systems[2] such as agriculture, industry, and transportation. Ubiquitous computing and new Artificial Intelligence (AI) methods, however, have enabled smart automation to become integral for (and in the process radically change) a growing number of everyday human activities. Smart AI-powered automated systems require enormous amounts of data, both to be developed (trained) and to carry out their tasks. Whereas "traditional" automation relied on a predefined set of instructions based on its builder's understanding of the process to be automated (e.g., steps in an assembly line), smart automation aims to be more flexible, reactive, and adaptive.

Smart automation based on Machine Learning (ML) adapts its responses based on patterns derived from vast amounts of information about human activities. Usually, the services provided by smart automation require predicting human behaviour, so the more data they have about a user's context and activities, the more effective their operation.[3] Hence why smart, automated systems are often deliberately conceived to acquire as much data as possible from their users—usually without their explicit knowledge and consent. Furthermore, given the value placed on data nowadays, some forms of automation are built for no other reason than to mine people's data. This continuous extraction and accumulation of information constitute the backbone of a new exploitative economic system threatening fundamental human liberties called surveillance capitalism (Zuboff 2015). Entire organisations are now invested in developing such "dishonest" forms of automation that not only extract data but, to do so, they lock users and operators into behaviours for which they have no legitimate need or desire (Girardin 2019).

As we have seen then, it is already virtually impossible to build error-free "traditional" automated systems because humans are integral to their operation. Any attempt to replace the human factor can trigger errors that will need to be addressed by *another* human supervisor (Strauch 2018). Nonetheless, the limitations and irremediable operational problems brought by this growing complexity tend to be solved with yet *more* sensors, more data, and more dependencies; that is, with more automation. This paradox arguably happens because those in charge of the design tend to be technically trained (and minded) people who are more concerned about machines and efficiency than the welfare of users and operators (Norman 2010). Often, their objectives involve attempts to quantify the unquantifiable and, in doing so, they bring more burdens to the very people whose work should be made easier by automation. This kind of design that is driven by technological capabilities and not by user's needs is, by all means, one that follows a *wrong* design approach.

3. Good Design is Humanist Design

Asking what is a “good” design approach necessarily implies asking what *is* “good” design first. As anyone acquainted with the field of design knows, there is no single, consensual, and absolute answer for neither of those questions. The reason is, mainly, that philosophical issues (i.e., problems open to reasonable disagreement) such as these cannot be addressed in absolute terms (Floridi 2013). First, they need to be properly contextualised, that is, constrained within an appropriate level of abstraction (LoA)—the epistemic interface that mediates our relationship with any given phenomenon (Floridi 2008).

The LoA here chosen starts by broadly characterising design as “the intentional solution of a problem, by the creation of plans for a new sort of thing” (Parsons 2015, sec. 1.1); then, by framing design as a quintessentially modern enterprise. First, because design, as a domain and as a discipline emerged more or less between the late eighteenth and early twentieth century, a period commonly described as the modern era (Erlhoff 2008). Secondly (and consequently), because it is in the modern era that the quest for characterising and systematising “good design” started taking shape. Thirdly, because it is in the modern era that the broader cultural, political, economic, and social consequences of “good” design was first noticed and problematised. Finally, because it is the modern era that regarded good design as the activity that could most aptly respond to the social challenges brought by industrialisation (Parsons 2015).

The modernist view of design, despite its excesses, continues to be a necessary reference, not only for historical reasons but because it assumes that good design, however it might be defined, must always pursue genuinely human(ist) interests. The LoA here chosen thus assumes that a good approach to designing must always start from human needs. This is, in general terms, the tenet of Human-centered Design (HCU)—also known as User-centered Design (UCD),[4] the dominant contemporary paradigm in design methodology—particularly in the context of interaction design (IxD) and user experience design (UX).

Nominally, UCD has its origins in the early 1980s, in the multidisciplinary Project on Human-Machine Interaction from the Institute for Cognitive Science at the University of California, San Diego, headed by Don Norman. The insights of the project were condensed in the influential book, *User Centered System Design*[5] (1986). Several of the ideas developed by Norman’s group echoed the guidelines proposed earlier by Gould and Lewis (1985) in the article “Designing for usability”, which made a strong case for adopting an empirical approach to systems design based on thorough user research and intensive cycles of prototyping and testing. These guidelines, in turn, can arguably be traced back to Scandinavian design approaches, in particular, to work developed by the Swedish Home Research Institute (*Hemmens Forskningsinstitut*, HFI) in the mid-1940s.

4. Looking closely at these two approaches, there are differences between them: while both are concerned with human perception and design, UCD may be regarded as a more compact subset of HCD. For the sake of simplicity, in this paper, they will be treated as interchangeable.

5. 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.

The HFI's main goal was understanding and improving the conditions and practices of housework through design research, but their work eventually led to the development of a full-fledged design methodology (Göransdotter and Redström 2018). Another Scandinavian approach with a robust User-centered ethos is Participatory Design, whose origins date back to the 1970s when researchers turned to ethnographic methods, action research, and a constructivist understanding of technology to tackle problems brought by post-industrial shifts. The goal of Participatory Design was empowering workers by allowing them to better transition to and deal with increasingly automated work environments by incorporating their tacit knowledge of manufacturing processes into automated systems development (Spinuzzi 2005).

Both the HFI's and Participatory Design methodologies contain more or less all the elements that characterise UCD frameworks as we now understand them. Which, to paraphrase Göransdotter and Redström (2018) include: starting by analysing user practices; employing interdisciplinary research; combining qualitative and quantitative data; carrying out iterative prototyping and testing at various levels of fidelity; regarding the user as an expert possessing crucial (tacit) practical knowledge; and involving different stakeholders throughout the design process. Furthermore, as Göransdotter and Redström (2018) also note, these approaches defended the need to always keep notions such as justice, inclusion, and user representation at the centre of the design process.

It follows that the aims and concerns of contemporary design methodologies, as well as the so-called "ethnographic turn" that has been shaping design since the early 2000s (see Blauvelt 2007), have deep historical, social, and political roots. UCD methodologies are far from homogeneous and are continuously evolving. However, at their core, they are all committed to the idea that technologies, particularly those involving computational (and therefore smart) automation, should be designed to improve people's lives. What this improvement means depends on the particular technology, the context where it is implemented, and the tasks it replaces, nonetheless a common aspect shared by all systems and promoted by UCD is that the designer's job is to make the complexity of the system accessible and, above all, *understandable* for the user. It follows that when correctly employed, UCD has a strong epistemological duty to end-users.

4. User-Centered Design and Complexity

Software-based systems are necessarily complex.[6] Complexity is present in (and arguably intrinsic to) many aspects of human life. Complexity is not problematic per se; it only becomes an issue when the user does not understand it and thus feels confused. Poorly understood complexity is, in this sense, *complicated*. Good User-centered Design is not about making things less complex (i.e., simple), but less complicated (i.e., understandable) (Norman 2010).

6. Complexity understood as the quality of something consisting of many interconnected parts and whose behaviour requires considerable amounts of information to be described (Bar-Yam 1997).

Whereas certain objects might be simple, the way we use them can be highly complex, and vice versa. To do what they do or to be used technologies generally *need* complexity. A violin is less complex than a wristwatch but playing the violin is undoubtedly a more complex task than reading the time. Software-based devices are complex objects that are comparatively more difficult to use their non-software based counterparts. A word processor is significantly more complicated to use (not to mention master) than any typewriter. Mechanical devices are usually more intuitive to use thanks to their "affordances"—i.e., the possible ways in which they can be used—are limited and readily apparent thanks to "signifiers"—the visible indicators for their appropriate intended use (2010, 227–28). Software-based devices are also more challenging to use because they have a broader range of context-dependent states. A hammer, for example, does not have modifier keys to alter its behaviour. In contrast, gestures in a touch-screen and commands in a keyboard can trigger a multitude of actions depending on the software being used.

The fact that using certain devices is complicated is not due to their inherent complexity but to poorly managed affordances. When that is the case, users feel helpless, powerless, and frustrated, since the behaviour of the system appears arbitrary and incomprehensible (Norman 2010). According to UCD, designers should strive to achieve a proper balance between the complexity of the underlying structure, behaviour, and limitations of a system, and the ways users conceive it. In other words, designers need to find a balance between the "implementation model" (i.e., the actual system's logic) and the "user's model" through a less complicated, usable "represented model" of the system (Cooper et al. [1995] 2014). Properly designed represented models (e.g., a usable and useful interface) empower users by making the complexity of the system reasonable, excusable, appropriate, and learnable. Taming complexity by designing an accurately represented model is not simple, however. Doing it implies dealing with a zero-sum situation.

Every technological system has "an inherent amount of irreducible complexity" (Norman 2010, 46). When a designer creates a less complicated interaction for users, the underlying complexity of the system increases accordingly. This paradox is known as "Tesler's law of the conservation of complexity" (Norman 2010; see also Saffer 2010, 136). When the interface is reasonably usable, the backend is likely complex; conversely, when the backend is optimised for the system's benefit, the user will likely have to deal with the complications of a confusing implementation model.

Design in this sense is about making otherwise complex systems accessible to people; it is about mediating the relationship between them and their technologies. The former implies taking a position about the role that artefacts play in human lives and anticipating the consequences that might emerge since arguably, every design is ultimately an argument about how people should lead their lives (Buchanan 2001).

When smart automation enters the picture, the consequences of "bad" implementations grow exponentially. Smart, automated systems are already complex networks of algorithms, sensors, human agents interacting under changing contexts at different time scales (Woods 2016). The inherent ironies of automation are further increased by the risks and complexities of contemporary computational infrastructure, namely, the fact that even the most simple interconnected system is embedded within a growing ecosystem of "balkanised operating systems, stacks of numerous protocols, versions, frameworks, and other packages of reusable code" (Girardin 2019). UCD should not only about making this complexity understandable but given how many aspects of people's well-being now depend on these systems, should also be about *protecting* people from the unintentional and deliberate misuses of these systems. In this sense, designers can no longer be just mediators between humans and technologies but also, or rather, mainly, "gatekeepers" and advocates of their users' interests (Monteiro 2019).

5. Some Principles for Good Design Now

From a methodological standpoint, UCD should be one of the best means to keep the consequences of poorly implemented automation at bay. Nevertheless, to do so, UCD cannot continue to be a mere epistemic facilitator, i.e., a means to make the complexities of technological systems understandable. UCD needs more than being empathic towards the user; it needs to assume a clear ethical stance towards technological development and, arguably, dare to enforce that stance. UCD might be the dominant paradigm in Interaction Design and User Experience design, however, our world continues to be further populated by dishonest forms of automation that incite users with "perverse incentives" (Loh and Misselhorn 2018) and dark patterns (Monteiro 2019). UCD needs to broaden its influence.

Given the way UCD-friendly approaches are being promoted nowadays it would be easy to infer that designers genuinely want to create things that make people's lives better; but either the methodology is not working, it is not indeed implemented, or it merely has become another form of white-washing—much like the better part of AI ethics. As shown by the recent scandal involving Joichi Ito and the MIT Media Lab, academic AI ethics has been easily manipulated into becoming a power-less bureaucratic seal of approval for otherwise unethical forms of automation (Hao 2019; Ochigame 2019) mostly because it has developed around the idea of voluntary compliance. Furthermore, designing an embedded Asimov-style moral code into smart, automated systems is a considerably difficult task for it involves reducing the complexities of ethical principles to procedural steps (Ceglowski 2016). Academic committees counselling on potential ethical mishaps with no actual influence over the design process is a recipe for inaction. Hence, it is up to designers to take the lead. It is the design process that should be imbued with an ethical framework.

The responsibility for protecting users' interests should be put on the designer, as well as in the product of her work. Ethics cannot continue to be regarded as an obstacle but as an integral aspect of doing design. Such ethical path might involve adopting a starker and more careful attitude towards automation, it should include a kind of *via negativa*[7] approach that allows designers to focus not on what could be beneficial, but on everything that could go wrong should a given form of automation is allowed to come into fruition. "Tire kicking" (Monteiro 2019) and testing for "brittleness" (Woods 2016) are helpful approaches in this regard. However, under certain circumstances, following an ethically minded UCD approach might involve doing away with the very notion of a minimum viable product based solely on functionalist standards.

7. That is, "the principle that we know what is wrong with more clarity than what is right, and that knowledge grows by subtraction" (Taleb 2018, Book I); see also Taleb (2012 Book VI) for a more detailed description.

An ethically-minded UCD approach should start by asking what the purpose of a given technological system is, e.g., asking whether its implementation will extend or hinder human capabilities. The former implies considering whether a given process should be automated at all. Automation, as we have seen, is an attempt to reduce the complexities of a given circumstance into piece-meal deterministic steps. However, because many aspects of any given process cannot be automated there is always room for unknown complications to emerge. These complications are further increased because, the "training" of smart automation is always done with known information, this narrowing exponentially increments the space of unknowns when dealing with complex situations, as the problem of induction illustrates. Rather than focusing on the "how's" of smart automation, an ethically-minded UCD should always begin with the "why's"; it should ask what kind of trade-offs should be assumed in the name of supposedly unobtrusive devices.

An existing tenet of UCD is that technologies should help people to make sense of the world better. For an ethically-minded UCD, this should imply making systems that are more transparent and honest about their limitations (a principle that was advocated by the well-known industrial designer Dieter Rams), potential side effects, and dependencies. Such a principle is already present in other fields; for example, the way drugs and medications are developed and regulated. This implies the system should be obtrusive under certain circumstances, more visible more honest about the "frictions" it might bring to its users. An ethically-minded UCD should always analyse tasks and goals within a broader context. Technologies are never neutral or inert, their introduction *always* affects a given context, their being and meaning are always situated, and the very fact of using a given instrument continually modifies the task itself. An ethically-minded UCD should be keenly aware of the mediation role of automated systems and technologies at large. This implies a broader understanding of "experience" and perhaps even eliminating the barrier between aesthetics and ethics, as these two domains may be, in fact, interlocked (Callard 2019).

Understanding automated systems as situated also means adopting a socio-technical systems approach to design. An ethically-minded UCD hence requires awareness of designer's blindspots about technology, that is, about the hidden assumptions they have about what a given system can and should do for users.

6. Concluding Remarks and Future Work

This paper began with the assumption that lousy automation is "bad design". It argued that good design should make complexity understandable, but that this sometimes involves assuming that specific processes cannot and should not be automated. As we have seen then, it is virtually impossible to build error-free "traditional" automated systems because humans are integral to their operation. Good design is not just empathy; good design is research, iteration, questioning, tire-kicking, and increasingly, an understanding of diversity. This paper has shown there is a lack of correspondence between the ideals of UCD and what in fact, comes out of design processes. It argues that what UCD should do is provide a richer, fairer, ethically sound, understanding of the world, but to do so, UCD needs to make explicit its stance on technology, it has to critically look at the moral, political and social consequences of our devices.

UCD should recognise that design is political in the broad sense. This realisation should be the starting point to determine what is good design and how that measure will affect design in the years to come. This paper, however, clearly has not "solved" UCD once and for all, nor has it proven beyond doubt what good design should mean nowadays. There is still much work to do in this regard. However, it has managed to start calling into question the perceived neutrality of UCD and show why it needs to be reformulated. A good starting point to do so is recognising that design nowadays is a collective enterprise that requires designers to work not on behalf of people but *with* people, and this means they should behave as gatekeepers with "skin in the game", who must strive to achieve not minimum viable, functional prototypes, but minimum ethical ones. Ultimately, designers should ask what role technologies play in developing human life; they should question themselves about what our human project for the future is.

Acknowledgements: This study was supported by UNIDCOM under a Grant by the Fundação para a Ciência e Tecnologia (FCT) No. UIDB/DES/00711/2020 attributed to UNIDCOM/IADE – Unidade de Investigação em Design e Comunicação, Av. D. Carlos I, 4, 1200-649, Lisbon, Portugal.

References

Autor, David.
2014. "Polanyi's Paradox and the Shape of Employment Growth." *The National Bureau of Economic Research*. September 2014. http://dx.doi.org/10.3386/w20485.

Bainbridge, Lisanne.
1983. "Ironies of Automation." *Automatica* 19 (6): 775–79. https://doi.org/10.1016/0005-1098(83)90046-8.

Bar-Yam, Yaneer.
1997. *Dynamics of Complex Systems*. Studies in Nonlinearity. Massachusetts: Addison Wesley.

Blauvelt, Andrew.
2007. "Design's Ethnographic Turn." *Design Observer*. May 1, 2007. https://designobserver.com/article.php?id=5467.

Buchanan, Richard.
2001. "Design and the New Rhetoric: Productive Arts in the Philosophy of Culture." *Philosophy and Rhetoric* 34 (3): 183–206. https://doi.org/10.1353/par.2001.0012.

Callard, Agnes.
2019. "Do You Want My Garbage?" The Point Magazine. August 7, 2019. https://thepointmag.com/examined-life/do-you-want-my-garbage-agnes-callard/.

Ceglowski, Maciej.
2016. "Superintelligence: The Idea That Eats Smart People." *Idle Words*. October 29, 2016. http://idlewords.com/talks/superintelligence.htm.

Cooper, Alan, Robert Reimann, David Cronin, and Christopher Noessel. (1995)
2014. *About Face: The Essentials of Interaction Design*. 4th ed. Hoboken: John Wiley & Sons.

Danaher, John.
2018. "Toward an Ethics of AI Assistants: An Initial Framework." *Philosophy & Technology* 31 (4): 629–53. https://doi.org/10.1007/s13347-018-0317-3.

Erlhoff, Michael.
2008. "Modernity." In *Design Dictionary: Perspectives on Design Terminology: Perspectives on Design Terminology*, edited by Michael Erlhoff and Tim Marshall, 262–66. Board of International Research in Design. Basel; Boston; Berlin: Birkhäuser.

Floridi, Luciano.
2013. "What Is a Philosophical Question?" *Metaphilosophy* 44 (3): 195–221. https://doi.org/10.1111/meta.12035 10.1111/meta.12035.

———. 2008. "The Method of Levels of Abstraction." *Minds and Machines* 18 (3): 303–29. https://doi.org/10.1007/s11023-008-9113-7.

Girardin, Fabien.
2019. "When Automation Bites Back." *Near Future Laboratory*. January 16, 2019. http://blog.nearfuturelaboratory.com/2019/01/16/when-automation-bites-back/.

Gould, John D., and Clayton Lewis.
1985. "Designing for Usability: Key Principles and What Designers Think." *Communications of the ACM* 28 (3): 300–311. https://doi.org/10.1145/3166.3170.

Göransdotter, Maria, and Johan Redström.
2018. "Design Methods and Critical Historiography: An Example from Swedish User-Centered Design." *Design Issues* 34 (2): 20–30. https://doi.org/10.1162/desi_a_00483.

Graeber, David.
2018. *Bullshit Jobs: A Theory*. New York: Simon & Schuster.

Hao, Karen.
2019. "In 2020, Let's Stop AI Ethics-Washing and Actually Do Something." *MIT Technology Review*. December 27, 2019. https://www.technologyreview.com/s/614992/ai-ethics-washing-time-to-act/.

Johnson, Deborah G., and Mario Verdicchio.
2017. "Reframing AI Discourse." *Minds and Machines* 27 (4): 575–90. https://doi.org/10.1007/s11023-017-9417-6.

Loh, Wulf, and Catrin Misselhorn.
2018. "Autonomous Driving and Perverse Incentives." *Philosophy & Technology* 32 (4): 575–90. https://doi.org/10.1007/s13347-018-0322-6.

Martinho-Truswell, Antone.
2018. "To Automate Is Human." *Aeon*. February 13, 2018. https://aeon.co/essays/the-offloading-ape-the-human-is-the-beast-that-automates.

Monteiro, Mike.
2019. *Ruined by Design: How Designers Destroyed the World, and What We Can Do to Fix It*. San Francisco, California: Mule Design.

Norman, Donald A.
2010. *Living with Complexity*. Cambridge, Massachusetts; London, England: MIT Press.

Norman, Donald A., and Stephen W. Draper, eds.
1986. *User Centered System Design: New Perspectives on Human–Computer Interaction*. New Jersey; London: Lawrence Erlbaum Associates.

Ochigame, Rodrigo.
2019. "The Invention of "Ethical AI": How Big Tech Manipulates Academia to Avoid Regulation." *The Intercept*. https://theintercept.com/2019/12/20/mit-ethical-ai-artificial-intelligence/.

Parsons, Glenn.
2015. *The Philosophy of Design*. Cambridge; Malden: Polity Press.

Saffer, Dan.
2010. *Designing for Interaction: Creating Innovative Applications and Devices*. 2nd ed. Voices That Matter. San Francisco: New Riders.

Spinuzzi, Clay.
2005. "The Methodology of Participatory Design." *Technical Communication* 52 (2): 163–74.

Strauch, Barry.
2018. "Ironies of Automation: Still Unresolved After All These Years." *IEEE Transactions on Human-Machine Systems* 48 (5): 419–33. https://doi.org/10.1109/thms.2017.2732506.

Taleb, Nassim Nicholas.
2012. *Antifragile: Things That Gain from Disorder*. New York: Random House.

———. 2018. *Skin in the Game: Hidden Asymmetries in Daily Life*. New York: Random House.

Vermaas, Pieter, Peter Kroes, Ibo van de Poel, Maarten Franssen, and Wybo Houkes.
2011. *A Philosophy of Technology: From Technical Artefacts to Sociotechnical Systems*. Vol. 6. Synthesis Lectures on Engineers, Technology and Society. Morgan & Claypool Publishers LLC. https://doi.org/10.2200/s00321ed1v01y201012ets014.

Woods, David D.
2016. "The Risks of Autonomy." *Journal of Cognitive Engineering and Decision Making* 10 (2): 131–33. https://doi.org/10.1177/1555343416653562.

Zuboff, Shoshana.
2015. "Big Other: Surveillance Capitalism and the Prospects of an Information Civilization." *Journal of Information Technology* 30 (1): 75–89. https://doi.org/10.1057/jit.2015.5.