# Insolvent

## How to Reorient Computing for Just Sustainability

**By:** Christoph Becker

## Citation:

*Insolvent: How to Reorient Computing for Just Sustainability*
**By:** Christoph Becker
**DOI:** 10.7551/mitpress/14668.001.0001
**ISBN (electronic):** 9780262374668
**Publisher:** The MIT Press
**Published:** 2023

The MIT Press

# INTRODUCTION
## JUST SUSTAINABILITY DESIGN

A systems approach begins when first we see the world through the eyes of another.
—Churchman (1979a, 231)

It took a plague to make some of the people realize that things could change.
—Butler (1993)

### NOT ALL IS WELL IN COMPUTING

In April 2022, a single bitcoin transaction produced over a thousand kilograms of $CO_2$ emissions and 350 grams of electronic waste. These numbers are enormous when compared to conventional payment systems like credit cards: the carbon footprint of a single purchase paid in bitcoin equals that of over 2.6 *million* credit card transactions (Digiconomist 2022). This is not coincidence: it is a systemic consequence of a key design decision in the bitcoin system that leverages the relative strength of compute power as a mechanism to decide what is valid (Nakamoto 2008). The largest pool of processing power decides, and the power consumption of the network as a whole is bound to rise together with the economic valuation of the currency: in 2021 alone, the carbon footprint of bitcoin transactions tripled. If its use continues, its emissions *alone* could cause global warming in excess of 2°C (Mora et al. 2018). The impact of this warming, and of other human-driven decline of nature, falls disproportionately on the world's poorest, who do not use bitcoin at all, and on nonhuman nature. One million animal and plant species are now on the fast track to extinction (Díaz et al. 2019).

The deep entanglement of information technology (IT) with our societies has raised high hopes for a transition to more sustainable and just societies. In principle, computing can be key to environmental sustainability and social justice. It can enable open information access, support sustainable lifestyles, drive the dematerialization of goods and services, and support the partial decoupling of value creation from material resource consumption. But in practice, the opposite is more common. Everywhere on this planet, computational systems drive environmental damage, increase the demand for resource extraction, reinforce inequality and injustice, and enable the erosion of privacy and democratic governance. This is especially true for those systems that are touted as success stories.

The computational systems we live with today often advance economic objectives at the expense of social and environmental considerations. Those we design tomorrow must advance social and environmental values simultaneously with technical and economic objectives. This fact has prompted many in computing to work on designing for sustainability and to tackle questions of justice and fairness in computing. But their good intentions have not changed the trajectory of computing. On the contrary: the rising tide of reports about the harmful implications of computing indicate that its debts are mounting. But these debts are treated as what economists call *externalities*—they are outsourced, offshored, foisted on those distant from the design processes.[1] From designed obsolescence to excessive energy consumption, from racist algorithms to the indirect implications of platform economies and surveillance capitalism, the choices made in the design and development of computational systems have implications at farther distances than ever before.

These choices come in many forms. I use the term *systems design* to refer to the broad set of intentional activities that directly or indirectly shape computational systems.[2] This encompasses activities such as software development that are often seen as "technical," and activities such as the facilitation of workshops with stakeholders that are often seen as "social." This simplistic distinction between technical and social hides that most systems design activities are simultaneously social and technical because they explicitly or implicitly reconcile social aspirations, expectations, needs, and concerns with technical constraints and material conditions. This is openly acknowledged in areas such as user interface design, architectural design, requirements engineering, or project management. Because of its wide sociotechnical spectrum, systems design is taught and practiced in a range of disciplines including human–computer interaction (HCI) and software engineering by many stakeholders, not all of whom consider themselves computing professionals.

The social and temporal distance between design choices and their outcomes entails uncertainty, ambiguity, and what moral philosopher Gardiner (2014) terms *asymmetric vulnerability*: those affected by technology development and design decisions have little means of influencing the outcomes—in the case of future stakeholders, none at all. One might say that these stakeholders carry the debts of computing: the wide-ranging ways in which the outcomes of systems design weigh on our planet and its societies.

Once we account for these debts, the question is whether computing can pay them back or whether we should consider it *insolvent*. As the title suggests, I will argue that in its current form, computing is indeed insolvent: It is incapable of paying back the debts it owes to this planet and its societies. It is so stuck in its ways of thinking that, to make progress, we need to rethink and restructure it.

This book will start from the premise that the rise of computing's debts is shaped by the ways of thinking that dominate it. This book's central argument is that computing can only pay them back and become a force for sustainability and justice once we rewrite central narratives of systems design. Three ideas in particular have shaped the discourse of computing like a river's undercurrents—that algorithms and computational systems are politically neutral tools; that problems and requirements are objectively given facts; and that the individuals and teams who design and develop computational systems are rational agents whose reasoning processes are admittedly flawed and biased and incomplete but, nevertheless, best described and approximated using the computational metaphor of information processing. The modern ideas of technological neutrality, scientific objectivity, and rational decision-making have played central roles in shaping the self-understanding of computing throughout the past seven decades, even though their validity and footing vary across the range of fields and perspectives in computing. In systems design practice and research, these ideas manifest *as myths*—"foundational narratives that are ritualistically circulated within groups to reinforce collective beliefs" (Ames 2019, 18)—when their limited scope is over-extended into a context in which their validity is not empirically or theoretically supported. Together, these myths drive a narrow focus that views design as using computational tools to solve problems described by scientific reasoning, based on the assumption that the resulting technology is value neutral. This narrow ideology is stuck. To overcome its impasse, we need to rewrite the narratives it is built on.

This book examines the role and validity of the myths of computing in order to reorient systems design practice and theory in computing toward the values of sustainability and justice. My aim is to illustrate *how* these ideas manifest in systems design *as myths*; to demonstrate how the myths interact to produce misleading beliefs about the nature and implications of systems design practice; to show how these misleading beliefs prevent meaningful engagement with the challenging questions raised by social justice and sustainability; and on that basis, to develop concrete steps to reorient systems design practice, research, and education.

In other words, to genuinely progress toward sustainability and justice, we need to combine the conversation about how computing shapes our lives with a close and critical look at how it comes to life—how we design computational systems and how we should design them. Doing so shows that despite many promising efforts, computing in its current orthodox form is not paying the dues it owes to our planet and its societies because its dominant mode of engagement through computational problem solving cannot recognize and reckon with the politics and values that shape technology design. The politics of technology design, the social dynamics of participation in design, and the cognitive factors of engineering and design decisions reveal how the myths of computing distort the discourse and shape what we can talk about when we talk about sustainability and justice.

## WHAT IS JUST SUSTAINABILITY DESIGN?

In this book, I introduce just sustainability design (JSD) as a framework for systems design practice, research, and pedagogy that privileges sustainability and justice and, therefore, the asymmetric and uneven effects of systems design choices at a distance. The purpose of JSD is to bring about improvement, not just avoid damage. Just sustainability design is sensitive to the discursive nature of systems design and the dangers presented by misleading narratives that keep systems design practice captive to a false consciousness. Because it centers the concerns of sustainability and justice, JSD must attend to five important factors they raise:

First, most profound effects that we recognize as sustainability or justice do not happen right away. Instead, they take place later, over longer time periods, removed from the moment in time and place where the decisions took place that caused them. So JSD must account for the uneven *dispersal* of design effects and their ripple effects across spatial and temporal scales.

Second, because the dispersed effects are distant, fragmented and scattered across these scales, they are *uncertain as well as ambiguous*. Uncertainty captures the understanding that there is a probability associated with these effects: few are certain, but some are more likely than others. Ambiguity captures the uncertainty about uncertainty: for many potential outcomes, we will not have probability distributions, and for many, their meaning to those affected is not immediately clear to the designers. Ambiguous outcomes can be interpreted in more than one way. These two aspects are related but distinct because addressing uncertainty is very different from addressing ambiguity.

Third, the many stakeholders—people involved in or affected by systems design activities, and non-human life affected by it—are not all able to participate equally, and future stakeholders are unable to participate at all. This causes a *fragmentation* of perspective and agency across the many stakeholders involved in or affected by design decisions.

Fourth, some stakeholders will have the ability and interest to influence systems design to various degrees; others will be affected indirectly and have much less influence, even if they are interested. This causes *power dynamics* across direct and indirect stakeholders with unevenly distributed influence over these decisions in systems design.

Finally, the nature of these complex ripple effects, and the fragmentation of perspectives, means that multiple interpretations of the same situation or outcome may be articulated based on assumptions that are mutually incompatible. Similarly, two stakeholders may evaluate outcomes in ways that cannot be reconciled easily, because there is no common scale of measurement. JSD must account for this *incommensurability* across different views of the complex design space it aims to address.

These five factors—dispersal, uncertainty and ambiguity, fragmentation, power dynamics, and incommensurability—raise questions that are social, cognitive, and political as well as technical. To address them, we need a theoretical framework for systems design in the twenty-first century guided by the criteria outlined in broad strokes below.

1. In a world that is deeply unsustainable and unjust, the uncritical pursuit of technology design as instrument in the unreflective service of dominant interests cannot possibly be a force for sustainability and justice. But we still need to build technology that makes sustainable and just societies a reality. Critique alone, as important as it is, will not bring about the constructive approach still required to do that. So JSD must be simultaneously *constructive and critical*: that is, it needs to build positive change while constantly questioning hegemonic norms and assumptions.
2. Because technology is social and society is technological, JSD must prioritize the integrated understanding of sociotechnical systems and their environments over the isolated analysis of individual components. In one word, it must be *systemic*.
3. Because sustainability and justice involve such a broad range of dispersed, fragmented, uneven, and sometimes incommensurable perspectives, JSD must represent a *pluralist* ground on which divergent worldviews can meet without requiring full consensus.
4. Because just sustainability is about the future, JSD must consider the *temporal* scale and dynamics of design in a historical context.
5. Technology never happens in a social or cultural vacuum. Far from being "independent of culture," JSD must be aware that its *perspective* will always be partial, built on unspoken assumptions, and it must be equipped to reflect on its already given context, boundaries, and assumptions.
6. In situations of ambiguity and incommensurability, goals of optimization are only appropriate or meaningful in a very limited way, as we will

see. The primary concern of JSD must be whether its proposed interventions are *legitimate*, rather than whether they are optimal.

These directions for JSD emerged through collaborative research in which I explored the role of computational systems in sustainability and developed tools that support the collaborative exploration of distant effects in systems design among heterogenous stakeholders (Penzenstadler et al. 2018; Becker et al. 2016). This work, discussed in chapter 1, allowed me to recognize the limitations of conventional systems approaches applied to sustainability and justice, as explored throughout part I. The principles of JSD are expanded and refined in chapter 9.

In its commitments, JSD is consistent with critical approaches to design such as design justice (Costanza-Chock 2020) and data feminism (D'Ignazio and Klein 2020). It shares its commitment to be constructive and critical with critical making (DiSalvo 2014; Ratto 2011), feminist HCI (Bardzell 2010), and crip technoscience (Hamraie and Fritsch 2019). It shares the aim to find paths off the rationalist highway with Rosner (2018) and its embrace of epistemic pluralism with Escobar (2018). What distinguishes it in orientation from these allies is an agenda of social change that aims to orient systems design practice toward the values of sustainability and justice by introducing a *critically systemic* perspective and approach to reflection, emancipation, and critique into the engineering methods of systems design in computer science and cognate fields.

## REORIENTING SYSTEMS DESIGN TO SUSTAINABILITY AND JUSTICE

There is no need to belabor the urgency of environmental sustainability in the age of the climate crisis. Others before me have mapped out some of the roles that IT can play in addressing sustainability (Tomlinson 2010) and highlighted the problematic role that IT also plays in fostering environmental destruction (e.g., Hilty and Aebischer 2015b; Crawford 2021). It is important to add the long-standing argument that environmental sustainability should never be treated in isolation from social justice (Agyeman, Bullard, and Evans 2003; Agyeman et al. 2016; Agyeman 2013), but the role of IT in social justice has been explored more competently by others

(Benjamin 2019; Costanza-Chock 2020; Eubanks 2018; S. Noble 2018). What can a book on sustainability, justice, and IT add to the above?

To my knowledge, this is the first book that explicitly focuses on the intersection of sustainability and justice in the context of systems design and explores the consequences. In doing so, I am less interested in adding substantively to the existing forceful critique of computing's material implications in systemic racism and inequity, or to the substantive exploration of the environmental effects of IT production and consumption. Instead, I focus on how issues of sustainability and justice get discursively recognized and addressed in systems design practice, research, and education. I am especially interested in how the myths of computing cause these concerns to be recognized, framed, and interpreted in particular ways, and I want to examine how conducive these framings are to genuine improvement.

Three examples will illustrate how the myths distort the discussions in subtle ways. A few years ago, I attended a week-long workshop on modeling for sustainability at Dagstuhl, the prestigious informatics center in Germany. Three dozen carefully selected experts, leading representatives of the fields of model engineering and sustainability in computer science, came together on invitation of the workshop chairs to discuss the role that modeling research should play in supporting environmental sustainability. The seminar room was full of critical thinkers, highly educated computer scientists who want computing to help the world transition to a sustainable way of life. At one point in the middle of the week, we agreed that we needed to discuss the intricate role that human values play *in modeling*. A breakout group formed to map out and discuss the thorny questions that arise. Or so I thought. After half an hour, the conversation took a turn, I took a backseat, and a mere twenty minutes later, the group felt they had completed the task. They had *modeled human values*. A neat chalk figure on a blackboard documented a decomposition of values into constituent elements and outlined a number of relations that would have to be represented in a model of human values. Our subject matter had been thoroughly defeated. Modeling had once again remained victorious. I must have had an incredulous look on my face when I asked if no one else thought that we had gotten off track? I was certainly met with bewildered looks—there was a palpable sense of disconnection in the room. What was I talking about? Wasn't this what we set out to do?

I struggled to explain that I was hoping to discuss how human values operate *in* modeling. What I and one other colleague had argued for was different in nature—we wanted to examine how deeply held beliefs and commitments (values) play out in social processes where human beings create models to reason about environmental sustainability.[3] We wanted a conversation that combined the constructive view, how model engineering can help to orient our life toward sustainability, with a close and critical look at how these models themselves come to life. That week is when I resolved to write what became this book.

Many in computing want our field to be a force for good—a force for sustainability, equity, diversity, justice, and similar values. Many are involved in projects with lofty aims along these lines, and many of these projects are worth pursuing. But more often than not, their work does little to make the world a more sustainable or just place. On an aggregate level, computing's environmental impact continues to worsen, and the reports of harmful technology and its implications in the erosion of our democratic governance systems continue unabated. At the time of the workshop, I had been working on sustainability in and through software systems for over a decade. I had shifted from a focus on the sustainability of digital software-dependent resources and information systems to a broader concern with the longer-term social and environmental implications of software technology design. Four years before the workshop, I had cofounded the Karlskrona Initiative for Sustainability Design. Its manifesto, discussed in chapter 1, calls attention to the need to consider sustainability holistically in every software project. It had received widespread attention and support in software engineering and requirements engineering. But a nagging suspicion had grown in me. I worried that the energy, momentum, and support for sustainability in computing were getting misdirected, seeing that the consequences and actions drawn from our arguments did not appear to enact lasting, transformative, sustainable change—merely business-as-usual under a new name.

The incident at the workshop brought home one central reason why. My colleagues did not consider how models embed values. Instead, they placed the arguments for the importance of human values in modeling into the conceptual and narrative structures arising from their disciplinary ways of thinking—above all, computational thinking. Inside that frame, the issue of human values *in* modeling becomes a challenge *of* modeling. Our question was meant to be a critical one: How are the resulting models shaped by the values of those who model? But this question is inaccessible to computational thinking and off limits to a reasoning mode of deductive logic in which the correspondence of modeled artifacts to "the real world" is taken as the only validity criterion of epistemology and then bracketed off from debate as an a priori condition not worth debating (B. C. Smith 1993). The question didn't even have to be explicitly reformulated—it was directly processed as a question of modeling. Why? Because the discursive worlds of the computing professions make it appear that way. They continue to lay out narratives that tell a story of computing and software technology as politically neutral tools built to address objectively given problems in processes best described as "problem solving" by groups of individuals whose cognitive processes are viewed as "information processing."

This was not the first time I had run against this brick wall. A year earlier, some of the same experts met at a different venue to discuss the same topic. One of the first presentations, presented by a civil engineer with environmental sustainability expertise, argued for *sustainability engineering* as an approach to "the *wicked problem* of sustainability." Early on, a slide with the ten properties of wicked problems. On the next slide: *How to solve this wicked problem*. Sustainability engineering was meant to be "objective; repeatable; sensitive to, but independent of culture; universal; and complete." I was baffled because, in my mind, the exact opposite was required: An approach that was not objective but dialectic; not repeatable but replicable; not independent of culture but embedded *with* culture, acknowledging the inevitable entanglement as a founding condition; not universalist but situated and contingent; and certainly not complete, but proudly incomplete and evolving. What was presented in this session as "systems thinking" really wasn't. It was deeply reductionist and failed to appreciate the nature of wicked problems (Rittel and Webber 1973). I scribbled furiously into my notebook.

Nor was it the last. A year after the values modeling exercise, I was back at Dagstuhl on a retreat. For the duration of one week, during lunchtime, I was seated with the participants of a Dagstuhl seminar on blockchains. By that time, I had started work on this book, and when asked about it, I spoke about it in terms of social justice and sustainability. Inevitably, some colleagues from formal areas of computer science would struggle to see how that was "computer science." Others would nod instantly and say that they also worked on this topic—usually, they had large amounts of grant money for some variation of "artificial intelligence (AI) for good." In one case, a grant on AI and blockchain for justice resulted in the worst lunch conversation I had in a very long time. It turned out that "justice," in the eyes of this eminent member of the distributed databases research community, meant an equal distribution of measurable risks and benefits among a set of individuals out of an identified population. I suggested that this narrow version of "fairness" is not the same as justice and that social scientists and others had something substantive to say about that topic. For example, we can hardly bring about justice merely by *defining* supposedly ideal institutions for justice and assuming that people's behavior will be compliant with those (Sen 2009), and it is crucial to distinguish between equal distribution of goods, equitable treatment of people, and justice. Algorithmic justice goes far beyond debiased fairness to include a historical understanding of oppression and injustice (cf. Costanza-Chock 2020, 63–65). But my comments were brushed off. As Jacobs and Wallach (2021) wrote later, the operationalization of essentially contested concepts of fairness and justice into narrow observable attributes is "appealing to computer scientists because they operate within the boundaries of a single computational system, without reference to the broader societal context in which the system is situated . . . these operationalizations necessarily lack aspects of the substantive nature of fairness" (382). This was not long after I had written the commentary for a brilliant polemic demonstrating just how inadequate computational reasoning mechanisms are when it comes to fairness, justice, and ethics (Keyes, Hutson, and Durbin 2019). I left the lunch a bit shaken. It was just too real.

These are not isolated incidents. I single them out here to illustrate how the undercurrent of the computing field shapes its discourse about sustainability and justice concerns. In hindsight, we can recognize in these stories the contours of the myths of computing. In the first instance, the meaning and texture of "human values in computing" were lost in a modeling view that considered only the problem of how to model values and failed to see that modeling itself is subject to values. Because the technology of model engineering itself was seen as a neutral tool, the idea that values might play a role in the construction of its artifacts was not even considered a possibility. Instead, the only question was how to use this tool to solve the problem at hand. In the second, the nature of sustainability as a wicked problem was misunderstood by an engineering view that interpreted it as a particularly difficult problem to be solved. And in the last, the nature of justice was interpreted through a computational lens that sees only a computational problem, unable to grasp what lies beyond that narrow field of vision. The individuals I encountered were not bad people, and they are brilliant thinkers in their fields. Like fish in water, my colleagues did not pay close attention to how their views were socially shaped and limited by the conceptual structures of their fields. Most fish in water have little appetite for learning how to walk, but fortunately, humans aren't fish, and many carry an enormous appetite for new grounds.

I examine four undercurrents and their interactions. In their naïve form, these are:

1. **The myth of value-neutral technology** tells a story of technology as neutral: that is, impartial and value-free. In this story, technology bears only facts. Values arise only in its interpretation.
2. **The myth of objective problems** tells a story of problems as objective entities that are the objects of problem solving and design. In this story, problems need to be represented correctly so that they can be solved using technology design.
3. **The myth of rational decision-making** tells a story of the human brain as an information processor making decisions. In this story, rationality is defined by reference to a normative ideal of decision-making embodied by a computer, and deviations from this ideal are treated as biases or mistakes. The story admits that the brain is imperfect but retains allegiance to the idea that rationality is the appropriate ideal in terms of which to understand it as an approximation.
4. **The myth of solvency** tells an optimistic story of technology as the savior solving problems for our world. In this story, the central activity of technology design is problem solving, and computer science is a problem-solving discipline. To design is to collaboratively solve

objective problems using value-neutral technology through a series of rational decisions. The story is headed to a happy ending. Unintended side effects are lamentable collateral damage, but they are exceeded by the benefit—it is "worth it." In using the terms *in/solvency*, I connect their common meaning—(in)solvency as the (in)ability to pay one's dues—to the notion of problem solving. A *solvent* in chemistry is also something that provides a solution.

To do justice to their texture and influence, this book will give space to each story, recognizing its merit, its scope of validity, and its historical evolution. It will develop a framework drawing from a range of fields I call the *critical friends of computing* to examine the limitations of each story and the distorting influence that it has on systems design for sustainability and justice. In more humorous terms, it will also sometimes add entries to a Devil's Dictionary of Computing.[4] Here are two:

> **Problem**, n.: something that can be fixed or solved.
> **Fix**, n. & v.: the source of tomorrow's problems.

## MY STANDPOINT

Two of these critical friends, feminism and critical systems thinking, emphasize the importance of clarifying the position from which knowledge is produced. Feminist standpoint theory speaks of embodied, *situated knowledges* (Haraway 1988). Since every perspective is partial, we need to understand in which way. Disclosing my own standpoint makes visible the partiality of my own perspective. Critical systems thinking speaks of implicit *reference systems* (Ulrich 1983), emphasizes the importance of making visible the boundaries and membership of what Midgley (2000) calls the "knowledge producing system," and to reflect on its positionality in the world by critiquing those boundaries. These concepts will return in chapter 5. For now, here is my standpoint.

I am a white cis man, born in Austria, of central European descent, with an invisible disability. I was born without that disability into relative privilege—a household of teachers and practicing musicians in Salzburg, a city defined by classical music and baroque history. I attended a conservative "humanist" high school where I learned Latin, Ancient Greek, and some Western philosophy. I read a book a day for most of my childhood; I played competitive chess and devoured chess books; and I learned to play the piano. My visible orientation as a nerd, combined with classroom dynamics and bad luck, resulted in permanent bullying throughout my school years. Chess opened a window to what I would later recognize as computational thinking, and I discovered, by way of chess, that I was really good at programming.

I was diagnosed with Type I diabetes at age seventeen, but interestingly, I never considered myself as living with a disability until much later, when I had lived through years of experiences of oppression through the North American sickness industry (pardon: healthcare system).[5] It took the lived experience of what Bowker and Star call the *torque* of classification systems (Bowker and Star 1999); of seeing my most intimate life signals monetized as data by "healthcare" startups; of lacking meaningful agency or control over the repurposing of my body's minute-by-minute statistics as data assets ("the new oil," remember?); of agonizing over endlessly repeated attempts to receive refunds for officially covered treatments from for-profit insurance providers; and of realizing just how much of my time is spent on managing my condition in ways able-bodied persons never have to see,[6] to recognize my situation as different. I am well aware that this experience is still vastly helped and buffered by my privilege as a tenured academic with private healthcare, and as a settler in Canada. Recognizing myself as such took me a long time (Lowman and Barker 2015).

My educational path to this position was supported by luck and a free public education system. This being central Europe, I never had a student loan or a credit score. Education was free, and healthcare was a given—a position of safety, freedom, and agency much more common in Europe than in the Global South or in North America. How privileged I was in comparison to others only became clear to me much later. My choice to study computer science opened ample opportunities to gain income and experience. Throughout my student years, I worked as freelance software developer, then software architect, then project manager, in projects of all sizes and various domains, from solo assignments on geodesic software to a contract as a software architect in the IT department of one of the world's largest financial corporations. This allowed me to develop a professional competence of delivering working technology that turned out to be invaluable later on. Only in part due to my work focus, I had terrible grades in the first few years of my bachelor's degree. But work experience taught me why I wanted the degree, and by the end of my first master's degree I had excellent grades. Yet I ended up in academia by serendipity. I hadn't applied for a doctoral program at a prestigious university, and I wasn't even thinking about it when I graduated in spring 2006. Instead, my thesis supervisor had a large European grant incoming and asked me to join it as a doctoral student and salaried full-time research assistant. I was mildly uninterested at first about the obscure subject of "digital preservation." I thought I belonged in the IT industry. I was convinced otherwise by a week-long summer school in a monastery in Tuscany, sharing breakout sessions in the garden with archivists and academics at the intersection of humanities and computing. Suddenly, I found myself in a group of people who genuinely cared about books. My doctoral program was free too, entirely unstructured, and formally detached from my grant-funded full-time employment as "project assistant."

I happened to thrive in this environment because I found myself able to deploy my project management and software engineering skills to deliver systems for decision support that helped shape a roadmap for digital preservation. This allowed me to win a European grant while graduating and a more foundational national research grant a bit later as a postdoctoral principal investigator. I realized that I loved the intellectual freedom of academic research. At that point, I had expertise in software technology, project management, and digital preservation, and I understood that "freedom" primarily as a way to find interesting problems to work on, get funding to do so, and pursue my intellectual curiosity in the process of solving those problems. Social responsibility or privilege were not yet a central part of understanding that freedom—nor was the possibility that some of these problems may have been "wicked." But I had also traveled widely; publishing articles in conference proceedings in various disciplines during my doctorate, I had the privilege of access to substantial travel funding. In the absence of care duties, I was able to append time for travel after many conferences. A formative side effect was a heightened appreciation of colonialism, inequality, social justice, and environmental destruction, as well as a growing awareness of the partiality of perspective, as I began to notice the blind spots of the European colonial tradition I inherited in education and through my social life in central Europe.

I undoubtedly had also subscribed to every myth identified in this book. The ideals of enlightenment rationality, computational thinking, problem solving, and IT as a savior were—implicitly—very prominent in my thinking. My Austrian-funded project included a peculiar research objective: "Longevity engineering" was motivated by the recognized lack of long-term thinking in systems design. The aim was to deliver metrics, measures, and evidence-based explanatory models for the lack of longevity in some systems but not others in order to integrate considerations of longevity into the lifecycle of information systems from the beginning (Proenca et al. 2013). A curious shift took place during this project. The failing attempt to quantify, measure, and model the "longevity" of systems so that it could be predicted and increased led me to recognize the framing as misguided. (From some different standpoints, of course, this would have been obvious all along.)

What began as an engineering-focused project led to *sustainability design*. It emphasizes a set of misperceptions about sustainability in systems design and proposes, in response, a set of commitments that make up the sustainability design approach (Karlskrona Initiative 2015; Becker et al. 2014). The double switch in terminology was significant: Instead of the engineering focus on measurement, prediction, and optimization, the manifesto places a focus on design—what Winograd and Flores (1986) call "the interaction of creation and understanding" (4); and instead of a *long life*, which can only be verified ex-post, it places a focus on the prospective "capacity to endure": sustainability (Fowler and Fowler 1995). The first small step toward the position in this book was taken.

In 2013, I joined the Faculty of Information at the University of Toronto on the tenure track. Thrown into a space where roughly a third of my colleagues hold a computer science doctorate, a third a social science doctorate, and a third a humanities doctorate, I began to read up on what seems like a self-replenishing Borgesian library department of all the books one must have read, and to appreciate the situated nature of knowledges. I faced the challenges of intercultural communication and some of the invisible systemic barriers that immigrants face in North America. As one of a few without a degree from what Toronto considers a "peer institution," the lack of a high-profile support network gave me some first-hand experience of the dubious mechanisms behind academic "meritocracy" (Labaree 2019; Iwen 2019; Appiah 2018) and the reactionary disciplinary politics of the Canadian peer-review system (Semeniuk 2017).

The unevenly distributed toll of the COVID-19 pandemic reminds me that despite the challenges I face as an immunocompromised person, I am incredibly lucky to be in a secure and privileged position. The vulnerability that comes with my disability has created a strange juxtaposition of privilege and oppression that has taught me to reflect and develop a certain sensitivity to privilege. I believe it also helps me to appreciate the nuance afforded by an intersectional approach that considers the *matrix of domination* (Hill Collins 1990).

As we learn simultaneously that the pandemic has dampened the accelerating rise of $CO_2$ emissions significantly, but only temporarily (Le Quéré et al. 2020), that the carbon footprint of Jeff Bezos's eleven-minute joyride in space exceeds the individual lifetime footprint of the planet's poorest billion people (Chancel et al. 2021), and that the shareholders of large tech companies are benefiting enormously from COVID-19 (N. Klein 2020), while poor communities, including those employed by Big Tech, were hit disproportionately hard in yet another illustration of systemic racism (K.-Y. Taylor 2020), it is clear that sustainability and justice are the challenges and opportunities of our lifetimes. Our societies *can* reshape significant parts of the way they operate. Each of us *has* "room for maneuver," to paraphrase Feenberg (2002)[7]—some have more, some have less.

Our future is in the hand of the collective actions of our societies. The design of computational systems plays one relatively central role in the reshaping of our ways of life. How we understand this role, and how we play our various parts, is our responsibility. This book aims to help us carry that responsibility through a systematic reflection on how we can think about systems design for sustainability and justice. It aims to help delineate the room for maneuver that exists for those participating in systems design, and writing this book is an expression of the room for maneuver I see for myself, from my standpoint.

Hindsight allows me to recognize the factors that stimulated my own appetite to recognize and transcend the water I was swimming in. My move from computer science to an information faculty entailed constant encounters with other disciplines, other norms, other values, and other literatures; in my research, the project of longevity engineering was going nowhere; and when the Karlskrona group committed to taking *systems thinking* as the transdisciplinary starting point for sustainability design in 2014 following compelling advocacy by Steve Easterbrook (2014a), I embarked on a chronological reading of systems thinking literature throughout the twentieth century.

## THE CRITICAL TURN IN SYSTEMS THINKING

This book draws significant inspiration from the work of critical systems thinkers. It uses their arguments to advance its own, and it shows some direct applications of their work to systems design practice. But I have learned that the broad term "systems thinking" is open to significant misinterpretations because it is used to reference an enormous variety of work that is typically classified in a wide range of disciplines. It is beyond the scope of this book to provide a comprehensive historical overview, but I want to retrace a few core concepts and developments.[8]

The core idea of Systems Thinking is the realization that some aspects of interest of the world cannot be decomposed into constituent segments without disappearing or losing their meaning: "to make sense of the complexity of the world, we need to look at it in terms of wholes and relationships rather than splitting it down into its parts and looking at each in isolation" (Ramage and Shipp 2009, 1). This idea, now commonplace, arose in many different fields—from biology to family therapy, from psychology to climate science, from ecology to pedagogy, from economics to organizational theory. Systems Thinking therefore comes in countless forms. Biologists Maturana and Varela explored the biological roots of cognition and knowledge and defined the concept of *autopoiesis* (Maturana 1980; Maturana and Varela 1992), while cultural anthropologist Margaret Mead thought about the dynamics of social change (Ramage and Shipp 2009). Economist Kenneth Boulding placed an understanding of economics into a comprehensive set of hierarchies and "considered the world variously as a physical, biological, social, economic, political, communication and evaluative system" (Ramage and Shipp 2009, 70), while professional practice scholar Donald Schön wrote about reflective practice in professions such as music and design. These approaches to systems thinking, to name just a few, have little in common in terms of their *content*. What they have in common is a way of thinking that pays primary attention to thinking about wholes and how they are constituted and organized in given contexts.

The proliferation of the "systems" term in colloquial language—the healthcare system, the education system, the transportation system, the information system—indicates that many fields are inherently looking at "systems" of education, management, ecology, public health, and so on. The century-long history of Western systems thinking has also resulted in systemic thought dissipating deep into the way we reason about the world. Few disciplines have *not* seen some form of systems thinking. Because of the diversity inherent in systems thinking, it can be difficult to understand what makes an approach *systemic*. Systems thinking often sits uneasy with disciplinary categories because disciplinary science "is essentially isolated by its disciplinary politics" (Churchman 1979b, 12) and therefore unable to engage with the full meaning of the world. In contrast to the "vertical" deep dive of investigations based on disciplinary theories, systems thinking approaches tend to transcend or even disregard disciplinary perspectives. The main reason is that disciplinary boundaries are rarely aligned with the phenomena of interest. Systems thinking therefore often favors a grounding in the empirical complexity and structure of the world. Churchman's systems approach to real-world planning—today, we would call it design—is motivated by the *environmental fallacy* (1979a). This fallacy describes an approach to solving an identified problem without regard for the environment it arises in and therefore without understanding the wider effects the intervention will have. The result, more often than not, is a situation that is worse than before. Instead of narrowing the perspective, a systems approach begins by expanding it. To understand what a bicycle is, for example, a systems approach will begin not by taking it apart but by asking what other systems it forms a part of. This may lead to an understanding of urban transport, commuting, and road safety.

Because systems thinking approaches have been built across such diverse fields out of heterogeneous ontological and epistemological commitments, they ultimately have very little in common other than the commitment to struggle with the aim for a "holistic" understanding of the issues they face.

Whatever the scale a particular systems approach emerges in—whether it is a living unit such as a cell as in biology, a cognitive unit such as a human person as in psychology, a classroom as in pedagogy, or a natural habitat as in ecology—the *system* concept "embodies the idea of a set of elements connected together which form a whole, this [whole] showing properties which are properties of the whole, rather than properties of its component parts" (Checkland 1981, 3). These properties are called *emergent properties*. The concept of emergence is relevant to all systems approaches, but they diverge wildly on how they conceptualize it.

Systems approaches also differ in how they organize the elements they are interested in. Some provide specific structural models of systems. For example, system dynamics considers measurable variables that represent structural properties of the world, such as the number of people currently alive or the amount of carbon dioxide emitted yearly, and organizes these variables with quantitative causal relationships that allow it to predict overall behavior as the emergent property. Soft systems methodology (SSM) considers purposeful human activities as its *element* and the *contingency* of each activity on other activities its main organizational relationship. Where system dynamics represents structural relationships between quantifiable variables as causal loops that can be expressed as equations (D. H. Meadows 2008), the purposeful activity models in SSM are graphs used to represent the contingency of goal-directed activities relevant to a social situation (Checkland 1981). Other approaches, such as the critical systems heuristics we will meet later (Ulrich 1983), do nothing of the sort.

Even within the approaches listed here, not all models are in fact models of reality. Many approaches do assume that their models represent elements or structural properties of the real world. They are referred to as "hard" systems approaches not because their systems are hard but because their epistemology is. For others, such as Checkland, systems are *ideas*: that is, mental or discursive constructs. SSM rejects the idea that the social world can be meaningfully represented in models. The purpose of activity models in SSM is not to model real activities but to develop a structure for a conversation about a social situation. And yet other approaches are agnostic to this distinction. Instead of debating whether claims about systems correspond to reality, they focus on how claims are discursively established. For example, critical systems heuristics focuses all its attention on boundary judgments as the central discursive act of design. These three kinds of epistemic assumptions are commonly used to distinguish between “hard,” “soft,” and “critical” systems thinking. A central concept that unites and simultaneously distinguishes all three groups of systems thinking approaches is whether a given system’s *environment* is treated as an unproblematic given, a central design choice, or a never-ending question.

JSD primarily relates to two types of systems approaches: system dynamics and critical systems thinking. System dynamics is a hard systems approach that underpinned the influential report *Limits to Growth* (D. L. Meadows and Club of Rome 1972) and which is central to various efforts to understanding, modeling, and predicting the human influence on the climate—so much so that its language has seeped into popular culture through concepts such as feedback loops, tipping points, carrying capacity, and leverage points. Within its conceptual framework, we can distinguish direct, indirect, and large-scale structural impact of systems design choices and explore nonlinear counterintuitive effects. It is within system dynamics that the concept of *leverage points*—places where small actions can cause large consequences—becomes operationalized into a very useful set of twelve categories (D. H. Meadows 1999) discussed in chapter 9. Its epistemology is structuralist—the assumption is that by modeling structural properties of the world, its future evolution can be predicted given a set of starting conditions. Because of its “hard” orientation, system dynamics has little to say about where its structures come from, where its assumptions come from, how to justify itself, and who gets to decide. Its direction is decidedly *analytic*—it proceeds by breaking down the whole system into constituent variables until they are considered sufficient to explain the observable behavior of interest, without being reflective or critical about the boundaries of the system of interest. In its ontological assumptions, system dynamics continues early systems thinking approaches that used terms like “systems theory,” “systems analysis,” or “systems engineering.” These early flavors of systems thinking have left a bitter taste: functionalist, imperialist, positivist, managerial, controlling, reductionist, instrumentalist. This is one reason to place these approaches in the context of the broader family of thought known as systems *thinking*. It is important to be aware that their shortcomings have been forcefully critiqued *within systems thinking*.[9] This realization is complicated somewhat by the coopting of the term "systems thinking" by prominent system dynamics proponents (Senge 1990; D. H. Meadows 2008). In systems approaches based on a positivist or structuralist epistemology, the systems idea often translates into a misleading belief that the systems approach is *comprehensive*. That belief can only be maintained in an analytic, inward-looking approach.

Critical systems thinkers maintain that a well-understood systems approach must begin by recognizing the tension between the desire to be comprehensive and the realization that it is impossible to be comprehensive (Ulrich 1983). As Churchman put it in the epigraph of this introduction, "a systems approach begins when first we see the world through the eyes of another." Shortly afterward, "the systems approach goes on to discover . . . that every world-view is terribly restricted" (Churchman 1979a, 4). *Critical Systems Thinking* opposes the purely analytic attitude that is prevalent among hard systems thinkers. It emphasizes a "synthetic" attitude, which maintains that things can only be truly understood once they are seen as embedded into their environment. Context matters in systems thinking, as it does in feminist thought (D'Ignazio and Klein 2020). It is this critical approach to systems thinking—reflective, emancipatory, and pluralist (Midgley 1996)—that will be of significant interest in this book.

In the 1960s, in rationalistic systems approaches such as operations research and organizational cybernetics, the scientific apparatus was brought to bear on social questions to shape societies at an unprecedented scale. Together with computing, these approaches had developed mechanisms of large-scale prediction and control (Churchman 1979a; Erickson et al. 2013). Critical systems thinkers grappled with the hubris of that project, with its epistemological challenges, and with its ethical implications. The critical turn they took, and the insights they developed on the way, remain invaluable today. Computing now is about to implement the project that the rationalistic systems approaches dreamt of in the 1960s, and it does so largely on the same epistemological foundations that underpinned rationalism then, despite well-articulated critiques (e.g., Winograd and Flores 1986). Whether we consider this utopian or dystopian, we have a lot to learn from this history that can help us orient systems design in the twenty-first century for sustainability and justice.

## THE AIM AND STRUCTURE OF THIS BOOK

This book makes the case for privileging the intersecting issues and values of sustainability and justice in systems design in computing and beyond, and it outlines an approach to do so. It argues that we must replace central narratives that have shaped the discourse of computational thought with more accurate narratives that are historically informed and provide a more nuanced set of metaphors to reorient the discussion about systems design. The ideas of value-neutral technology, objective problems, rational decision-making, and solvency form widely held but false beliefs underneath the discourse and will be treated as myths. While they are rarely taught as “facts,” and many in computing will laugh about them outright, they remain central to the orthodox rationalistic discourse of computer science. Just sustainability design provides methodological principles and commitments that help researchers, practitioners, and educators avoid the distorting influence of these myths and design for sustainability and justice. This book is designed to enable computing professionals and researchers to identify these myths, point to them, surface them in comparable situations, and thus chart paths around their gravity wells in research, education, and practice.

Part I poses and responds to the question: “Is computing able to pay back its debts to societies?” Chapter 1 sets the stage by laying out the role of computing in environmental sustainability. I survey the recognition of the lifecycle impacts of computing technology and the role of computing in improving environmental sustainability. I compare different paradigms of sustainability and sustainable development to highlight key challenges and developments in ICT for sustainability and sustainable HCI, and I describe the emerging paradigm of sustainability design. Chapter 2 expands this view to address social justice. It introduces the concept of *just sustainabilities* and the connections between sustainability and social justice. On this basis, it explores what I call the *debts of computing*—the widespread, often hidden, and usually indirect effects of software systems on their social, economic, personal, and natural environment. Chapter 3 examines the adequacy of computing’s primary forms of reasoning—computational thinking—to these challenges. It concludes that the current discourse of computing is shaped by the myths of objective problems, rational decision-making, and value-neutral technology. It outlines how these beliefs manifest in systems design *as myths*, and it demonstrates how these myths interact to produce misleading arguments about the nature and implications of systems design practice.

Chapter 4 shows how these three myths (value-neutral technology, rational decision-making, and objective problems) interact to form the fourth (solvency, or the idea that computational problem-solving makes the world a better place). Solvency is central to what I call *problemism*—a rationalist preoccupation with framing and solving problems. In computing, problemism occurs when the lens of computational thinking turns data-driven problem-solving into a tunnel vision, unaware of the social construction of problems and data, the varied forms of individual and social cognition and decision-making, and the politics of stakeholder engagement. A set of examples illustrates how these myths substantively constrain the direction of systems design activities and reinforce the unjust, unsustainable modes of current mainstream practice. The conclusion: the dominant discourse in computing is currently unable to address its debts to societies. Rather than to open bankruptcy proceedings, I suggest a restructuring of the narratives of systems design with help from neighboring and distant disciplines.

Part II introduces the critical friends of computing. With the help of their insights, it restructures the central narratives of systems design to facilitate the reorientation toward sustainability and justice. The conceptual basis for the central arguments is drawn from two areas that have a lot to offer computing: science and technology studies and critical systems thinking. These are introduced in chapter 5. The subsequent chapters address each myth in turn to examine where it came from, how it prevents meaningful engagement with the challenging questions raised by sustainability and justice, and how to replace or sidestep each myth.

Chapter 6 explores the myth of value-neutral technology. It contrasts it with insights from HCI, science and technology studies, and the philosophy of technology to show how values become facts in systems design. Chapter 7 explores the myth of rational decision-making in light of research from cognitive psychology and related areas. The conclusions suggest that a significant portion of behavioral research in systems design commits what I call the "normative fallacy"—it relies on normative frameworks to describe what people do, and thereby fails to provide reliable insights. This has significant implications on how we understand systems design practice, particularly how practitioners make decisions that have uncertain and ambiguous effects at a distance. Chapter 8 explores the myth of objective problems, drawing from a range of areas interpreted through critical systems thinking frameworks. Because problems are inevitably framings of particular aspects of a situation created by someone for a purpose, the politics of stakeholder engagement need to be made visible. That requires a critically systemic view on what it means to design in *wicked problem situations*.

The implications of this restructuring lie partly in practice, partly in education, and partly in research, and the chapters in part III explore each of these domains. Chapter 9 describes JSD as a coherent framework to systems design that is critical but engaged in productive design and engineering work. It then briefly introduces the idea of *leverage points* to organize possible avenues for change. The subsequent chapters present concrete methods and research projects that translate this methodological foundation into a research and design practice reoriented toward the values of sustainability and justice. Chapter 10 describes *critical requirements engineering*, an emerging approach to requirements engineering that combines its normative frameworks with critical systems heuristics, and it illustrates its application in a project. Chapter 11 develops new research directions in the cognitive dimensions of decision-making in systems design. It explores the implications of the *more-than-rational* view of judgment and decision-making in light of the challenge of psychologically distant effects, and it illustrates this new research direction with recent empirical studies on intertemporal choice. Chapter 12, finally, asks how this reorientation relates to professional competence. With a focus on social responsibility and collective organizing, it surveys the recent tide of initiatives for social change in computing and discusses the limitations of, and alternatives to, professional ethics codes in computing. The conclusion places the proposed reorientation of computing into a broader context of societal reorganization.

## AUDIENCE

This book is addressed to two audiences. First, I hope that it can help those within computing to orient their thinking, assess the role of implicit false narratives in their work, and position the angle of their work to avoid the torque of false narratives. For those who already focus on environmental sustainability and social justice, such (re)positioning will help them identify those leverage points they can act on to lift computing out of its unjust and unsustainable ways. Others may find new angles on their existing focus that help them to address issues with social import that currently resist transformative change. This applies to graduate and undergraduate students as well as researchers and professionals for whom JSD should provide useful guidance. For tech workers keen on social change, I hope the book provides arguments that can help them disarm what Cathy O'Neil called the "weapons of math destruction" and instead build "algorithms of liberation" (Roberts et al. 2018).

If you are a computer scientist who normally reads technically oriented papers, I thank you for your curiosity and I ask for your patience. This book will build a vocabulary beyond the terms common in fields like software engineering, and it will use that vocabulary to speak to and with these fields. In doing so, it will uproot some of the terms commonly used in computing to question what they really (should) mean. This will take some time.

Second, I write it for those in neighboring fields that are not focused on design or engineering—information studies, communication, media studies, social justice, environmental sustainability, science and technology studies (STS)—who collaborate with the more design- and engineering-focused computer scientists in the space of sustainability and justice in IT. My hope is that it may help them understand how exactly computational modes of reasoning can limit the discourse and perspectives of computer science, show how the perspectives of *their* disciplines can be brought to bear on the blind spots of computing to effectively offer a helping hand as a critical friend, and perhaps help them to identify and make visible other myths of systems design that have not been recognized as such.

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Stone Serif by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Becker, Christoph (Director of the Digital Curation Institute), author.
Title: Insolvent : how to reorient computing for just sustainability / Christoph Becker.
Description: Cambridge, Massachusetts : The MIT Press, [2023] | Includes bibliographical references and index.
Identifiers: LCCN 2022038283 (print) | LCCN 2022038284 (ebook) | ISBN 9780262545600 | ISBN 9780262374651 (epub) | ISBN 9780262374668 (pdf)
Subjects: LCSH: Electronic data processing—Social aspects. | Computer systems—Environmental aspects. | Information technology—Social aspects. | Sustainable development.
Classification: LCC QA76.9.C66 B435 2023 (print) | LCC QA76.9.C66 (ebook) | DDC 303.48/34—dc23/eng/20221121
LC record available at https://lccn.loc.gov/2022038283
LC ebook record available at https://lccn.loc.gov/2022038284