This is a section of doi:10.7551/mitpress/14668.001.0001

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

6

SOFTWARE IS NEVER NEUTRAL

HOW DO VALUES BECOME FACTS?

Designing computer artifacts is an inherently value-based activity, deeply implicated in longstanding political struggles of the wider society in which computer science is embedded. Rather than viewing this fact as a breakdown in what should be a disinterested project, this alternative position embraces the place of systems development as a critical arena for the expression and enhancement of values.
—Suchman (1998)

Through design, the values of those involved shape the artifacts of the future. But how exactly does this take place? Systems design and engineering have developed ways to become more sensitive to the role values play in design and to make productive use of our deliberative and reflective capacity to elicit, negotiate, and critique values. This work has drawn on a wide range of influences, including moral philosophy, ethics, and psychology. It may seem absurd to believe that technology could be value-neutral, but the myth of value-neutral technology remains strong even today (Ko et al. 2020; Shilton 2018). To ensure that the political nature of systems design is broadly recognized and addressed, we must articulate the role that values play in it.

There is an important distinction to be made between values and value conflicts on the one hand and ethical imperatives on the other hand. Broadly speaking, ethics focus on what is the right thing to do, while values point to what we consider important. In the current computing discourse, the term ethics carries a strong normative framing in prescribing how to decide what to do, exemplified in the ethics codes that dominate the conversation. On that basis, "ethics is framed as a problem that can eventually be 'solved'" (Metcalf, Moss, and boyd 2019, 8). The myth of value-neutral technology (VNT) supports the primacy of ethics codes as prescriptive rules to be followed in design and engineering because it suggests that following these rules is sufficient to be ethical.[1] In contrast, the term "values" points to a more fluid, descriptive accounting of fundamental beliefs and preferences that direct and orient individual and organizational work. This sensitivity is often absent from ethics-focused work. The prevalence of VNT may be related to the focus within engineering disciplines on normative views of ethics (Spiekermann 2016, 2) and practice (see chapter 7). Moving beyond it reveals a perspective of sensitive, reflective, and critical opportunity.

This chapter briefly traces some of the origins of VNT to examine where it comes from. It then conceptually examines *how values become facts* in systems design, identifying a range of disciplines that offer fragments of a puzzle which we can usefully piece together. This is followed by a brief discussion of what is left to do for values to ethically shape just sustainability design.

ETHICS, VALUES, FACTS, AND THE MYTH OF TECHNOLOGICAL NEUTRALITY

Evacuating values from technology design in favor of prescriptive ethics principles and rules has a long history, rooted in the question of value judgments in social, purpose-oriented decisions. Ulrich (1983) locates important steps toward what we now call VNT in the work of Hobbes, Weber, and Popper. A milestone is Weber's distinction between *means* and *ends*—supposedly, the latter result from value judgments (where do we want to go?) while the former can be picked purely on questions of facts (what is the best way to get there?).[2]

> **facts**, n. pl.: claims made about the environment of technology design for which those who make them forgot to question where they came from, how they came about, which values they embody, whose values these are, whose facts they are, and whose interest that serves.
>
> **assumptions**, n. pl: <u>facts</u> about the project at the time of <u>kickoff.</u>

Sound familiar? The false dichotomy between facts and values is the basis of the prevalent logic suggesting that engineers and technologists should simply figure out *how* to do things (means) while "someone else" should decide *what* to build (ends) (cf Shilton 2018, 151f). It is the logic used by some AI advocates to claim that facial recognition technology, autonomous weapon systems, or automated hiring systems, are just a neutral value-free means without politics. "It depends on how they are used," they say, without regard for the values already expressed in and reinforced by the technologies they advocate for. Their reasoning is dubious. In his defense of VNT, Pitt provides a seemingly logical dissection of Langdon Winner's (1980) classic example of the New York underpasses. Winner suggests that their low height, which prevented the public buses used by socioeconomically segregated Black communities from accessing Long Island Beach, manifests political values. Pitt claims that he is simply unable to locate any values in technical artifacts.

> Where would we see them? Let us say we have a schematic of an overpass in front of us. Please point to the place where we see the value. If you point to . . . a number signifying a distance from the highway to the bottom of the underpass [and] tell me that is Robert Moses' value, I will be most confused. There are lots of numbers in those blue prints. Are they all Moses' values or intentions? (Pitt 2014, 95)

The claim that the drawings merely represent technical facts and therefore *not* values is ultimately based on the same logic that positions the prescriptive rule systems of ethics codes as necessary and sufficient guarantees of ethical conduct. But the basis of this logic does not hold up to scrutiny.

> Here is the snag. Decisions on means are never questions of fact only, for all means are in need of critical examination with regard to the value implications they themselves contain . . . means and ends are not substantially distinct categories but rather different perspectives for considering hierarchies of goals: what appears as a means from "above" (from the next-higher system level in the goal hierarchy) appears as an end from "below". . . . Once this is clearly understood, it seems almost unbelievable how uncritically a majority of contemporary social scientists, led by the logical empiricists and critical rationalists, have adopted the dogma that means and ends are substantially distinct categories, so that only "ends" are supposed to involve value judgments while "means" are understood as value-neutral. (Ulrich 1983, 72)

So, while any technological system is of course a means to some ends, different perspectives will reveal the same system to be a different means to different ends, and more importantly, every system also carries value implications beyond those expressed in its purpose. In this light, Pitt's argument looks rather disingenuous. No one claimed that a distance value in an engineering drawing is an exact representation of a human value and expressed as such. Drawings that represent technical facts still *express* values by virtue of their relationship to the world they are in. It is precisely in these models, placed and interpreted within their social context that gives them meaning, that these values find their expression. If we are willing to decipher them, we might find it easier than expected. Let us add an entry to the *Devil's Dictionary*:

> **models**, n. pl. (computing): the carpets under which, if we look carefully, we can find the human values, politics and moral decisions that have become code, features, qualities, documentation and other technological facts through the social practice we call *systems design*.

WHICH VALUES BECOME FACTS IN SYSTEMS DESIGN

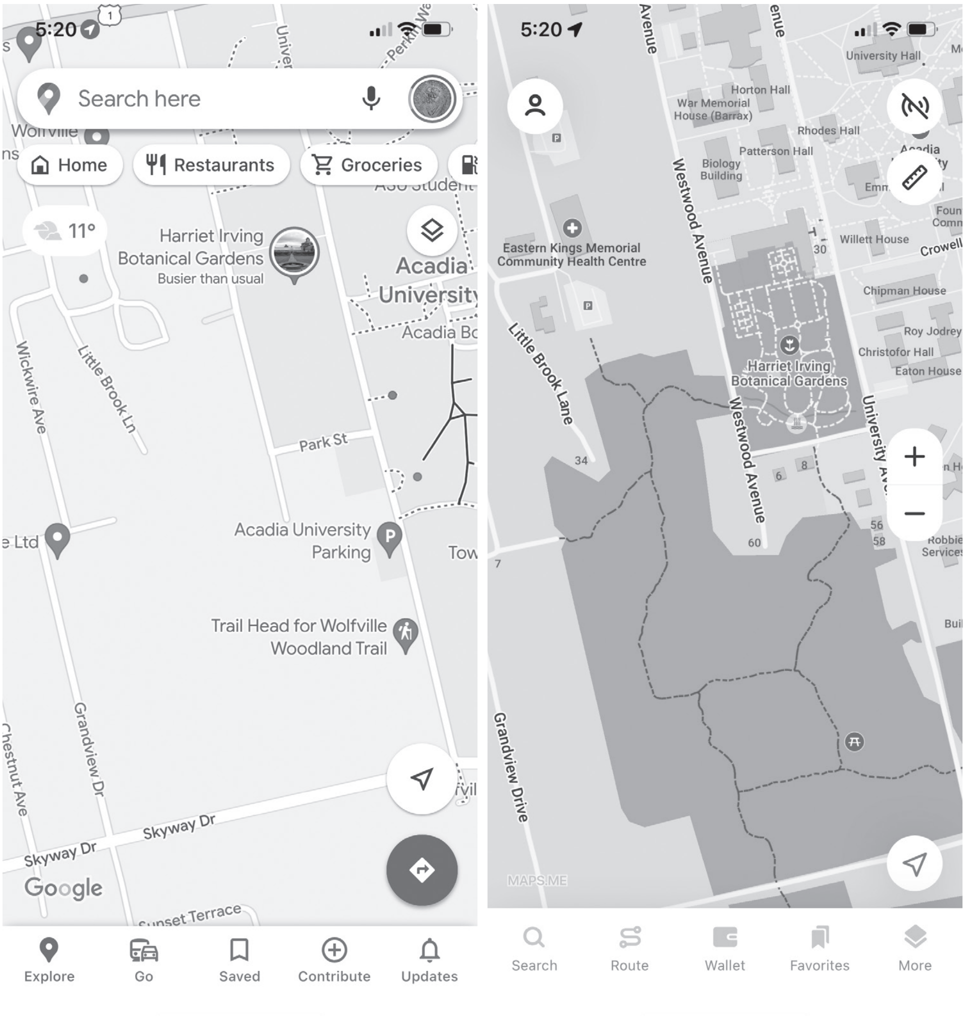
I propose a brief exercise. Open Google Maps, navigate to your favorite city, zoom to your favorite large park, and have a close look at what you see. Make a screenshot. Now look at the same place via OpenStreetMap. What do you see?

Maps are a quintessential example of scientific objectivity. We tend to think of the mapping techniques developed over millennia as a pinnacle of the objective representation of our world through technologically constructed artifacts. Scholars of cartography have of course long shown that mapping has never been a neutral technology. Maps are often drawn by victors, colonizers, settlers, missionaries, and oil companies. Maps have been drawn to mark territories, identify oil drilling sites, and support state surveillance and genocide. On the other hand, mapping technologies are also used by civil rights advocates and land defenders across the globe to fight injustices.[3] Even before adding objects, the projection of our globe onto a two-dimensional surface involves value-loaded choices about which distortions to introduce. Still, in our daily life, when we think of the content of maps, we typically evaluate it based on how "accurate" and "complete" we believe the content to be. In discussions of objectivity in science philosophy, values such as accuracy, correctness, and completeness are often referred to as *epistemic* values and admitted into the scientific process, while *contextual* values, the personal preferences of the scientist, are to be eliminated from the conduct of science and relegated to the choice of research question. This builds on Weber's distinction between means and ends (Reiss and Sprenger 2017; Diekmann and Peterson 2013).

On my recent vacation to the Canadian East Coast, I took a walk around the neighborhood of a friend's place in Wolfville, Nova Scotia, to bring down a high blood sugar. For driving directions, I use Google Maps, but for exploring places on foot, I turn to OpenStreetMap. There's a reason for that, illustrated in the screenshots of the same place, at the same time, rendered through these platforms (figure 6.1).

The two maps are both accurate representations showing important facts of Wolfville. But which values do they show? Google Maps takes a commercially oriented view. It shows us all the places where we can spend money. OpenStreetMap, the Wikipedia of maps, shows little of that in many regions, but look at the detail by which it represents public spaces. Every university building is named, every park bench is marked, and in your favorite park, I bet every drinking water fountain is marked too. For two platforms focused on mapping the features of our world, they show striking differences. The crucial difference stems from how they are disposed to different aspects of the world that we value. These maps manifest the organizations' and editors' diverging interests in different features of the mapped world and different purposes they choose to pursue.

Feenberg's concept of *formal bias* is useful here. It stands in contrast to subjective or organizational bias such as the commonly criticized kind that refers to a human bias based on content or substance such as gender, race, or ability. In contrast, formal bias is embedded in a rational form such as a technological platform or paradigm. For example, an automated hiring system trained on historical data may reinforce and amplify prior human bias manifested in the data, data generated by individuals who exhibited discriminatory patterns of hiring decisions that favored White men over Black women. The resulting algorithmic bias has become an important concern. "This sort of bias is properly called 'formal' because it does not violate formal norms such as control and efficiency under which technology is developed and employed" (Feenberg 2014, 166).



6.1 One place on two highly accurate maps (Google Maps, OpenStreetMap/maps.me).

But there are additional nuances of categories that we should distinguish. First, it matters not just where but how bias manifests. Formal bias can be easy to detect when it literally surfaces in the content or *substance* of technological platforms: OpenStreetMap shows different objects than Google Maps. When it resides in the invisible or intangible structures of affordances, features, qualities, and constraints, however, it is much harder to detect. For example, the OpenStreetMap API allows users to edit the

Table 6.1 Selected values prioritized by Google Maps and OpenStreetMap

	Explicitly prioritized	**Implicitly prioritized**
Values in Google Maps	social content	Commercial value Centralized control
Values in OpenStreetMap	distributed, peer-produced content	Objects in public space Shared editorship

map content, and many apps accessing the platform support this feature. Google provides much narrower paths for users to provide *consumer feedback* via reviews of places represented on the map. Via these affordances, each platform configures its users' roles. In similar ways, gig economy platforms such as delivery apps limit and constrain what each participant can do and see; automated hiring systems, beyond the evaluation and ranking, also structure the process of applications and the collection of data about jobs and applicants; and robotic automation technology structures the division of labor around it. When Amazon automates warehouse tech so that a machine controls and monitors how humans pick up objects, that tech embodies the removal of agency from the worker and subordinates them to the machinery (Crawford 2021, 48–74). In contrast, when a warehouse equips its workers with better machinery to help them pick up objects they could not personally carry for physical reasons, while leaving it to the workers to coordinate their activities, it embodies the amplification of physical strength and personal autonomy. Projects to design either type of system may have efficiency improvements as primary purpose, but other values also shape and constrain the systems' dispositions. Because these examples stay within the formal norms of conceptual frameworks such as "usability" and "features," some will consider this kind of system as "neutral" even when it marginalizes one group's values in favor of another group's values.

Second, it is fair to assume that formal bias is compliant with the local norms of a given context, but it can very well violate formal norms defined elsewhere. For example, substantive formal bias is addressed in norms of fairness and non-discrimination (to some limited degree), but these are not always implemented and enforced. In fact, organized advocacy for a change of norms is often directed toward making possible the local contestation of formal bias of substance or affordance under these new regulations.

In discussing values, the relevance of some is more readily accepted and barely doubted because they are culturally embedded into engineering norms. Values such as efficiency, effectiveness, and reliability are treated in engineering the way epistemic values are treated in science. Others, such as worker agency, may be much more contested. But in fact, there is no fundamental difference between these values and their influence on the design process. As Feenberg writes, “when the division of labor is technologically structured in such a manner as to doom subordinates to mechanical and repetitive tasks with no role in managing the larger framework of their work, their subordination is technologically embedded. Inequality is enforced by the very rationality of the machine” (Feenberg 2014, 166). The dismissal of “worker agency” as a “loaded value” is not in any way a neutral act. And none of the systems listed here are neutral.

HOW VALUES BECOME FACTS IN SYSTEMS DESIGN

Different systems, even if focused on the same overall purposes (such as mapping the world), will embody different values simply because they are the manifestations of different interests. These values manifest in subtle and less subtle ways through the range of features and qualities of systems. Can we describe more precisely how this happens?

Prior work on values in systems design provides us with useful, if partial, responses to this question. Significantly, it has mapped out how values as discriminating criteria in deliberative processes can be intentionally elicited and used to drive design activities to shape models and artifacts. Value-sensitive design (VSD) provides theory, tools, and methods that support designers in facilitating intentional conversations about specific values when they design artifacts. It is meant to sensitize those involved in design toward the role of values. It focuses on how values and the “moral imagination” can shape design outcomes (Friedman and Hendry 2019). It offers sensibilities to longer-term implications too via supporting justice-oriented systems design across generations, exploring “how the element of time might be leveraged in design processes with deep-seated value tensions” (Friedman, Nathan, and Yoo 2017, 83).

To do so, VSD proposes a trio of interacting perspectives: conceptual explorations, empirical investigations, and technical analyses are meant to proceed iteratively and in parallel. The relational concept of *values* *suitabilities* connects an artifact's specific technical property, such as a feature or an affordance, to a value such as worker agency. Thus, it supports what Brey called the "moral deciphering" of technical artifacts (Brey 2000; 2010). Technical investigations in VSD can be retroactive analyses deciphering existing artifacts or proactive forms of design. VSD researchers have also investigated which factors support the surfacing and centering of values conversations in design (Shilton 2012). VSD envisions these streams to be integrated, although critics have pointed out that some aspects of this integration remain underdeveloped (Manders-Huits 2011).

Value-based engineering (VBE) (Spiekermann 2017; Spiekermann and Winkler 2020; IEEE 2021), on the other hand, treats values in much the same way that requirements engineering treats requirements: as the object of applied scientific activity (Akkermans and Gordijn 2006). This introduces the idea and ideals of scientific objectivity into its methodological foundations. Values in this family of approaches are stakeholder positions that are elicited, documented, weighted, measured, and when in conflict, negotiated. In doing so, this work introduces the concept of *value disposition*: a system quality positively or negatively disposed toward an identified human value, much like an affordance in design is disposed toward a potential human action and parallel to a value suitability in VSD.[4]

There is an important difference between this objective, supposedly neutral stance of VBE and the formative, orientating stance of VSD. VSD's attention centers the designers as subjects, while VBE's focus lies in removing subjectivity by prescribing processes to treat values as objects of applied scientific measurement and construction. In VSD, it is a designer subject(ivity) who is *sensitive*, while VBE centers the supposedly objective value "base" of engineering. In the terms of critical systems heuristics, the guarantor for VSD is a designer; for VBE, it is method. In the spectrum between these two, we find approaches that use scientific measurement frameworks for values (Schwartz 1992; 1994) to drive software engineering and design processes (Whittle et al. 2019; Ferrario et al. 2016) or use *values* as a central concept in requirements engineering (Thew and Sutcliffe 2017).

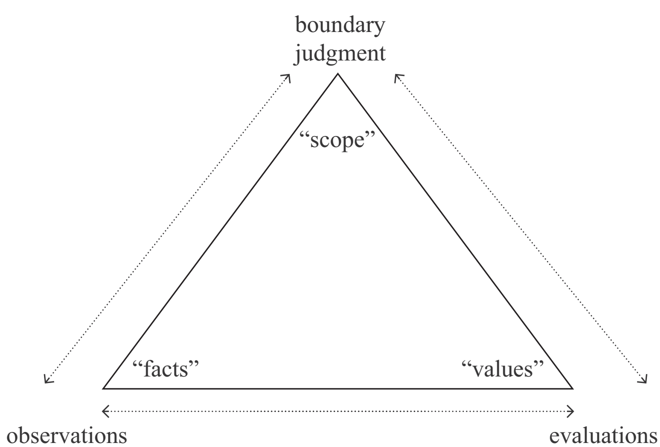
But this explicit articulation of how values are intentionally expressed by and impressed onto artifacts is only half of the story. Values (including value tensions) also *implicitly* shape the discourse and, directly or indirectly, models and artifacts—not only by being referenced as discriminating criteria in deliberative processes, but also by shaping the narrative and themes of conversation, thus *orienting* those who design. Values shape preferences and thus choices, including choices *about* which values to articulate. For example, a group of participants in a design project may use envisioning cards, a VSD tool (Friedman and Hendry 2012), as part of their value-sensitive design process. Within this process, however, values such as group conformity will influence which cards participants select, which values are given prominent space, and how the conversation is structured. Values will also shape participation itself and the articulation of evaluation criteria for design process and product. While VSD is better prepared to be sensitive to the nature of this shaping, neither VSD nor VBE incorporate a critical examination of *implicit value positions* and their influence in design. As Shilton writes,

> it is important to acknowledge the limits of a values-oriented approach to design. Attention to values and ethics within design can help to ensure that new technologies do not perpetuate or aggravate existing bias, or create new unfairness. But values-oriented approaches do not address larger power structures that perpetuate bias and unfairness in technology hiring practices, design education, or technology regulation. Values-oriented design methods are just one part of a larger culture of ethics that must become part of technology education, scholarship, and practice. (Shilton 2018, 150; see also Manders-Huits 2011, 282)

There are, however, approaches that can help us to examine the values and power structures beneath the surface. Computing's critical friends have at least three avenues to offer. First, through a range of approaches to discourse analysis, they have examined which values underpin computing *research fields* such as machine learning (Birhane et al. 2021), computer vision (Scheuerman, Denton, and Hanna 2021), and ICT4S (Knowles 2013). Second, the *mindscripting* method (Allhutter 2012) allows teams to deconstruct the stories they tell about their work in order to identify and better understand the identities, value positions, tensions, and power dynamics inherent in the situation they design in (Allhutter and Hofmann 2012). This is not a straightforward toolkit, but it has been used successfully to explore values and value tensions in computing spaces as distinct as machine learning (Allhutter and Berendt 2020) and requirements engineering (see chapter 10). Third, critical systems thinking (CST) methods support "closely examining the assumptions and values entering into actually existing systems designs or any proposals for a systems design" (M. Jackson 1991). This is precisely what is missing in the puzzle.

Drawing on discourse ethics, Ulrich's work considers the relationships between discursive claims about facts, values, and the scope of boundary judgments as illustrated in figure 6.2. In making boundary judgments as discussed in chapter 5, we observe and evaluate facts based on the reference system that scopes our assessment of what constitutes the relevant field of knowledge. Values shape the selection of facts, and vice versa. Recognizing new facts as relevant implies a reevaluation and rescoping of the system of interest. Similarly, shifting the scope of that system changes what we perceive as relevant facts and values (Ulrich 1998, 6; cf. JafariNaimi, Nathan, and Hargraves 2015, 97). Critical systems heuristics (CSH) can be used to make the values shaping the scope of the system visible; to allow those involved in design to reflect on their beliefs about purpose and improvement; and to create a space where those affected can (to some degree) question the values of those involved.

Similar relationships between facts, values, and boundary judgments hold in modeling (Diekmann and Peterson 2013). Contextual values, not just epistemic values, influence which properties are represented in models. In terms of figure 6.2, they influence boundary judgments and thereby facts; they influence the parameters used in models; they influence the choice of competing models; and sometimes, they are in outright conflict with epistemic values. For example, the *simplicity* of a design solution may conflict with a disposition to a substantive value such as *fairness*. In



6.2 Facts, values, and scope (adapted from Ulrich 1998, 6).

the values discussion mentioned in the Introduction to this book, these kinds of value influences could have been at the forefront. Instead, the attention of the audience centered exclusively on the visible *object* of values and applied a modeling lens *to* this object: we modeled values. The result: a collection of facts about values. Which values implicitly guided this discussion? How did they shape and constrain the selection of facts the group chose to represent in its model of values? Today, I would be better equipped to guide the group in that direction.

Ironically, this anecdote illustrates an important observation from CST: "the boundaries of accepted knowledge define the values that can emerge. Similarly, the values adopted will direct the drawing of boundaries that define the knowledge accepted as pertinent" (Midgley 1992, 9). The boundaries of knowledge that implicitly drove the conversation were the disciplinary boundaries of model engineering and software engineering. The crucial questions, however, fall squarely within the domain of computing's critical friends. The power dynamics within the group—which individuals were central, which were considered leaders in the field ostensibly being discussed, how the question may transcend that field, who was comfortable speaking up, and so on—interacted with epistemic norms and values of what constitutes a good model. Again, the triangle of observations, evaluations, and boundary judgments provides a lens into the mechanisms that elevate some stakeholders' knowledge claims and forces others' claims to the margins (Midgley 1992). These dynamics do not happen at random. Similarly, the authors of a recent machine learning study contrast dominant values with conflicting values that get silenced, based on meticulous deciphering of published literature: "computer vision datasets authors value efficiency at the expense of care; universality at the expense of contextuality; impartiality at the expense of positionality; and model work at the expense of data work" (Scheuerman, Denton, and Hanna 2021, 1).

In design, sacred and dominant values get impressed upon artifacts, while other values get repressed (marginalized in political pushback) and suppressed (relegated to profane status). Those values that in turn become impressed are often oppressive to those stakeholders whose values have been marginalized. This is a process of politics—the shaping of social organization—as much as it is a process of engineering (the systematic application of scientific procedures)—both at the same time. And it is why systems design is inherently a value-loaded activity. By deciphering which

Table 6.2 Examples for techniques to surface explicit and implicit values

	Proactively, during design	Retroactively, given an artifact
Explicit values	Value-sensitive design techniques such as *envisioning cards* help to orient design teams to values. *Value-based engineering* processes such as IEEE P7000 introduce some values concepts into the engineering vocabulary and support systematic mappings. *Value-based requirements engineering* similarly centers values, motivations and emotions in RE.	Technical investigations in VSD can be retroactively conducted to decipher the value suitabilities of existing artifacts (Friedman and Hendry 2019, 89). VBE techniques and models could in principle be applied retroactively.
Implicit values	*Critical systems heuristics* can be deployed in ideal mode to orient a team toward reflecting on implicit value positions (Ulrich and Reynolds 2010): see chapter 10. Comparison between ideal ("ought") and actual ("is") brings value conflicts to light. *Mindscripting* supports teams in surfacing values and negotiating value tensions (Allhutter and Hofmann 2012).	*Critical systems heuristics* deployed in reflective mode can unearth further value conflicts, and in polemic mode, it can make value suppression visible (e.g., Ulrich 1981; McCord and Becker 2019). *Deconstruction* more generally can bring to light the value commitments of engineering methods and artifacts (Allhutter 2012, Allhutter and Berendt 2020), and the contrast between espoused and manifested values (Beath and Orlikowski 1994).

values are expressed by the artifacts, we can make some of this visible in hindsight. But to proactively open the conversation, we must combine the complementary views brought on by computing's critical friends *during* systems design, as far as that is politically feasible. Table 6.2 summarizes some techniques that can be employed to help.

BEYOND NEUTRALITY: THE POLITICS OF SYSTEMS DESIGN FOR JUST SUSTAINABILITY

The asymmetric vulnerability and distance that characterize just sustainability design sharpen the need to become critically aware of the processes of marginalization by which values turn sacred or profane, and to incorporate perspectives on *distant values* proactively into systems design. On the explicit/proactive quadrant of table 6.2, envisioning cards provide a reasonable starting point, but, on their own, they are insufficient because asymmetric vulnerability creates a power differential that VSD does not know to address.

A strictly normative uncritical approach that prescribes rules for action cannot address uneven power relations nor close the gap between those who design and those who are affected. It will therefore do nothing to change *whose values* dominate and shape systems design. This means that we need to combine multiple methodologies in an approach that is both constructive and critical, both critical and systemic. When designers shape artifacts with a moral imagination sensitized to sustainability and justice, they must also critically interrogate their assumptions, value positions, and power. When engineering and design perspectives translate stakeholder perspectives into value dispositions and technical specifications, we must similarly combine them with a critical examination of the normative reference systems that underpin these translations. We need the critical friendships between engineering disciplines and computing's critical friends to figure out how to make this work.

In systems design practice, all this calls for a social change embodying a shift of priorities toward values and toward different values than those currently dominating computing (Barendregt et al. 2021). A systems design practice reoriented to just sustainability design will strive to incorporate value sensitivity considering all quadrants of table 6.2.

This shift is decidedly a political project that embraces "systems development as a critical arena for the expression and enhancement of values," as Suchman calls for in the chapter's epigraph. In terms of research, we need to examine how methods from the four quadrants can be combined. The explicit shaping of artifacts with a moral imagination sensitized to sustainability and justice must be complemented with a critical interrogation of assumptions, value positions, and power. In terms of education, we need to demonstrate more clearly than existing work "how values become facts" by building concrete and convincing cases that comprehensively document to engineering audiences the tangible intermediate steps that link explicit and implicit value positions to value dispositions and their material effects.

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