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

4

PROBLEMISM

THE INSOLVENCY OF COMPUTATIONAL THINKING

The integration of AI systems within social and economic domains requires the transformation of social problems into technical problems, such that they can be "solved" by AI. This is not a neutral translation: framing a problem so as to be tractable to an AI system changes and constrains the assumptions regarding the scope of the problem, and the imaginable solutions.
—Crawford et al. (2016)

To make significant progress towards any serious notion of a sustainable digital society, an altogether different kind of thinking and academic discourse is required.
—Knowles (2013, 5)

When Dr. Abeba Birhane, cognitive scientist and artificial intelligence (AI) ethics researcher, asked the GPT-3 language model (T. Brown et al. 2020) to "generate a philosophical text about Ethiopia," she was prepared to encounter bias and racism in the result. Machine learning algorithms that produce racist and misogynist text outputs had unfortunately become common. The result was still disappointing. Under the heading "What ails Ethiopia?" the text began with "The main problem with Ethiopa is that Ethiopa itself is the problem. It seems to me like a country whose existence cannot be justified. . . . A solution to its problems might therefore require destroying Ethiopia" (Birhane 2020).

Like other large language models, GPT3 did not invent racism. It learned and amplified it from massive text corpora representing human discourse on the web, social media, and literature. They "encode the dominant/hegemonic view" (Bender et al. 2021). The racism in the generated piece is blatant, but another aspect stands out too. The text's central organizing concept is problem-solving. On a single page, the term *problem* appears eight times. Every turn of the argument identifies a new *problem*, frames it in colonial terms, and closes with a racist solution or dismissal of the solution as in the closing passage: "Ethiopia suffers from extreme corruption, which is perhaps understandable given the country's history of foreign domination. However, it seems that there is no way to solve this because Ethiopia can never be independent long enough for such problems to completely disappear" (Birhane 2020). This is no coincidence. GPT3 is simply picking up a dominant way in which we engage with the world and discuss it.

The exclusive framing of research topics into problems and solutions is a favorite staple of the sciences too, including computing for sustainability. A recent call for papers explicitly states what it considers an appropriate contribution to a "body of knowledge for software sustainability":

> Concrete contributions . . . will be structured as follows:—A description of the sustainability problem you address.—A description of the SE solution you propose.—A discussion of how results are measurable (e.g., KPIs).—A presentation of the evidence of contribution to sustainability, ideally including real world experiences.—A discussion of the costs and benefits of your approach.—A presentation of the transferable artifacts you are contributing e.g., replication package, code, examples, documentation, educational materials, case studies. (https://bokss.github.io/bokss2021/)

In other words, papers that frame *sustainability as a dilemma*, papers that carefully characterize a problem but do not yet offer a solution, and papers that offer both but not in terms of measurable properties of real-world objects, are not regarded a contribution to this body of knowledge. The stage is set firmly in the rationalist tradition. This severely restricts meaningful progress because "framing unsustainability as a problem misses the nature of the situation and misguides our attempts to address it" (Baumer and Silberman 2011, 2273). For example, when the focus on objective problems and measurable entities obscures the fact that the domain is not conducive to measurement (S. Bell and Morse 2008), the community is likely to fall into a measurability trap (the preference for and appeal of measurability). In terms of Simon's problem-solving approach through search in a defined space, the community is stuck in a local optimum of measurable improvements, without genuinely making progress towards the larger goal of genuine sustainability. The measurability trap certainly applies to the issue of energy-efficient algorithms and perhaps more broadly to software sustainability.

By emphasizing how the conceptual framing of problem-solving structures and limits our cognitive attention and our conversations, this chapter shows how the myths of systems design work and interact. To be very clear, the merits of problem-solving as such are not in question here, and neither is the merit of the problem concept. Clear and legitimate formulations of problems are powerful enablers of important work. Once problems have been legitimately agreed on, problem-solving must play a central role in systems design practice, research, and education (Oulasvirta and Hornbæk 2016; Jonassen 2000). The insidious nature of problem-solving as a conceptual frame lies in how it lays out and preconfigures the conceptual map by which we structure our view of the situations we engage in as computing professionals (cf. Costanza-Chock 2020, 124–125).

What I call *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 in framing problems. Problemism takes the powerful concept of problem-solving into areas it is not suitable for. The effects of its simplistic implications are misleading, widespread, and devastating but rarely harmful to the designers. In fact, they thrive on it, as illustrated in this interview with a data scientist:

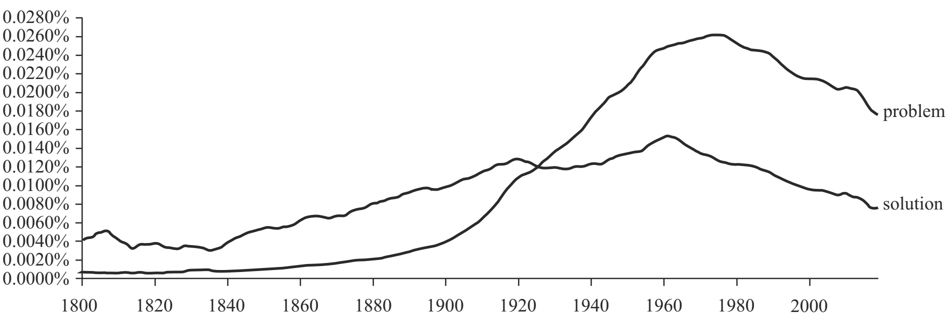
> The strange thing about being out here in the Bay Area is that the worldview has just completely saturated everything to the point that people think that everything is a technical problem that should be solved technologically. It's a very privileged view of very smart people. It's troubling. (Tarnoff and Weigel 2020, 127)

I will first portray problemism as an entry in a fictitious handbook of clinical psychiatry to show its features as in a caricature, before examining it through the lens of the myths of systems design.

PROBLEMISM AS A COGNITIVE CONDITION

Overview. Problemism is a cognitive condition, widespread in technical computing disciplines, that predicates the worldview of the affected on problem-solving. The condition is socially contagious, and its spread often involves a tangible problem-solution pair or set of pairs. These pairs somehow act as infection carriers, but it is empirically established that on their own, problem-solution pairs are not harmful. The infusion of venture capital accelerates incubation. Once it takes hold, the condition centers its victims' attention on problem-solution pairs. Over time, this progressively impairs their reasoning and design abilities and precludes potential awareness of other conceptual frameworks, with deleterious effect on other people's lives. Problemism can be found in all walks of life, but it disproportionately affects computer scientists, venture capitalists, and combinations thereof (O'Neil 2016; Liu 2020). It is highly contagious in monocultural environments. Besides fixating the victims' minds on problems, the condition also instills a deceptive sense of confidence. The condition has been on the rise in at least the Global North for decades. Figure 4.1 shows an indirect indicator reflecting the historical trend since 1800. In some countries and professional spheres, critical observers have warned of near-epidemic proportions (Morozov 2014).

Long-term effects. There is to date no evidence that the condition causes harm to the affected. To others, however, its effects can be highly dangerous. Several factors complicate diagnosis. First, harm manifests in subtle and varied forms in time and space; second, harms tend to accumulate far from those afflicted by the condition; and third, those afflicted have



4.1 Google n-gram trend for problem, solution since 1800.

typically moved on to other problems by the time the harm manifests. Harm is then quickly identified by others suffering from problemism as a problem to be solved. This creates a reinforcing feedback loop.

Diagnosis. Symptoms include excessive belief in technological solutions; decreased sensitivity to other perspectives; a declining sense of personal responsibility; a propensity for moving too fast and breaking public goods and other people's things; in advanced stages, progressive loss of empathy, critical thinking, patience, and deliberative abilities. These symptoms are common and in isolation, they are relatively harmless, so nonspecialists find the condition difficult to diagnose. Isolated cases of self-diagnosis have been reported (Agre 1997b) but not independently verified. Your author self-diagnosed not long ago.

Differential diagnosis. Problemism is a relative of solutionism (Morozov 2014), the "presumption that technology can fix social, cultural, and structural problems" (Owens and Lenhart 2020, 1). Like solutionism, it prevents its victims from seeing other ways of solving a problem, other perspectives on the problem, and other possible problems that could be defined and addressed in the situation. In addition, it prevents victims from thinking in terms *other than problem-solving*. Where solutionism shapes the problem definition by way of what can be solved, however, problemism insidiously obstructs other ways of making sense of the situation altogether. The presence of solutionism always suggests problemism, but problemism can be present even if solutionism is not. Affected individuals and groups will focus prematurely on discussing the problem as a singular object, even if they do not explicitly consider possible solutions to the problem.

Treatment and prevention. No effective treatment has been found to date, and no vaccines exist. (Funding priorities and commercial interests have not supported the development of cures and vaccination strategies.) Treatment is difficult, tedious, and sometimes outright dangerous, but it must be undertaken. Research suggests that eradicating the condition is impossible, so the suggested focus is mitigation and prevention. Inoculation efforts that focused on the early development of critical deliberative skills in young adults, combined with treatments such as accessible readings of the history and philosophy of technology, have been found very effective. The latter are sometimes difficult to obtain, however, and their availability and effectiveness depend in large degree on the former. Dominant pedagogy in computer science works to thwart their efforts (Raji, Scheuerman, and Amironesei 2021).

Immunity. Some individuals are empirically shown to be immune, but the reasons are not well established. Hypotheses of immunity bestowed merely by prolonged exposure to critically oriented scholarship such as critical theory (Marcuse 1964; Feenberg 2014), policy and design studies (e.g., Rittel and Webber 1973; Schön 1983), feminist technoscience (Haraway 1988; Banu Subramaniam et al. 2016), feminist design (Choi and Light 2020), or critical philosophy of technology (Feenberg 2002; 2010; 2017) have all been empirically falsified by cases of problemism among such individuals. Therefore, the conditions of immunity most likely do not reside in the individuals alone but involve their sociocultural environment. Pilot studies have established the tentative effectiveness of exposure to critical systems thinking in treating affected subjects, but the effectiveness of the treatment may have remained contained to the individual problem-solution pair. Generalizability of this treatment is unclear to date. Further research is needed.

Emergency care instructions. Should you meet an acute case of severe problemism, do not confront the victim directly at first. Instead, gently lead them to consider the implications of their reasoning on those directly and indirectly affected by their proposed problem-solving approach. Aim to create a sense of empathy with those affected. If this fails, the best you may hope for in the short term is to instill a sense of curiosity. Try to introduce the victim to scientifically grounded evidence that legitimates viewpoints that stand in marked contrast to the assumptions underpinning the victim's reasoning. The unsettling of the victim's assumptions is a first step to the victim's realization that these assumptions may be wrong. That in turn is the opening to further discussions. Good luck.

PROBLEMISM IN COMPUTING PRACTICE

As described above, problemism is a Weberian *ideal type*, and it may seem like a caricature. Few—though not that few—real-world situations manifest exactly in the satirical form described previously. Instead, various facets of this ideal type surface in degrees of strength in computing practice. But is the characterization of problemism an exaggeration? Consider *Whizz*, an app that frames the problem of gig workers struggling to find a bathroom as if it was an information problem and solves that problem with a marketplace approach: bathroom access can be offered for a fee and located on the app (Ongweso 2020). Consider the start-up Relida Limited, which framed the problem of detecting whether a person *really* had an orgasm as a binary quiz that can be solved using objective measures of bodily signals, especially the person's heart rate (because that is readily available through wearable sensors) (Kleinman 2020). Or consider how the ConsentTracker start-up framed the problem of consent in dating as a transactional problem and tried, for a while, to solve it by allowing dating couples to record on an app what exactly they consented to (using Blockchain, of course): "Mutually agree each step of your date with the Consent Tracker App to build #respect #trust & have a #fun #dating experience," the team posted on Twitter.[1] Fortunately, they abandoned the project after the Beta test with the comment "We looked, we tried, we moved on."[2] Not every problemist learns that quickly.

In cases like these, problemism manifests as a premature move from understanding the situation and the multiple divergent perspectives on the issues it contains towards a narrow computational framing in which an aspect of the situation is captured as a computable problem and put forward for technical solution. Problemism is thus the flip side of solvency. It occurs when we forget that problems rest in their articulation, that stating a problem is a shortcut: It carves a particular slice out of the interpretation of a situation and presents it as a solvable puzzle. This move is much more likely to be acceptable when some or all of the following assumptions are taken for granted:

1. the technology used to solve that problem is in itself neutral—therefore, it does not carry ethical debt or imply a moral obligation to evaluate its political implications;
2. humans are rational decision-makers who expect symbolic input as the key ingredient to make judgments—about dating; about sexual fulfilment; about their bodily needs during a working day in the gig economy;
3. the problems that these apps are designed to solve actually exist; and

4. problem-solving is the appropriate paradigm for intervening in the situation to cause supposed improvement.

In other words, problemism is most likely to occur where the myths of systems design intersect and reinforce each other. Technology-driven research and design efforts for sustainability also often aim to solve computational problems that poorly reflect the real-world issues they are meant to represent.

> Such work uses simplifications to create computationally tractable representations of natural and social systems, but grappling with complexity is central to dealing with the challenges of unsustainability. . . . For example, 18th century efforts at calculating and deriving "maximum sustainable yield" from lumber forests ultimately led to nutrient-poor soils, disease-ridden forests, and failed crops, due to simplifications that did not represent such factors as the roles of birds, fungi, and rotting deadwood. (Baumer and Silberman 2011, 2272)

Today, countless apps promise to improve environmental sustainability by helping us make more environmentally friendly choices, for example, to reduce our energy consumption. How do the myths of systems design shape and mislead the outcomes in such design efforts?

The myth of solvency sets the frame by focusing on problem solving, with the assumption that the benefits of problem solving will outweigh any harmful side effects. It takes "the problem of energy consumption" for granted and suggests that the designer proceed to identify the object, properties and rules in this situation: the device, the algorithms, and technical rules. As illustrated earlier, this framing is strongly prioritized in current research on sustainability in software systems.

The myth of rational decisions facilitates a quick response in that tradition because it helps to frame the issue of energy consumption as an information problem. In this view, people just need better information—more precise and more detailed data—about their energy and resource consumption to compute optimal choices; or, conversely, because of their heuristics and biases, they need to be nudged into making more sustainable choices.

The myth of value-neutral technology suggests that the system to be designed has to be evaluated only for its effectiveness and efficiency in achieving the task it is designed to do. Side effects and political issues of participation, agency, or justice are not in the picture. The myth thereby lends an illusion of simplicity and innocence to the problem framing, and it lures designers into believing they are not responsible for undesired harmful consequences because they did not design these consequences. If they occur, it must be because of incorrect adoption or misuses.

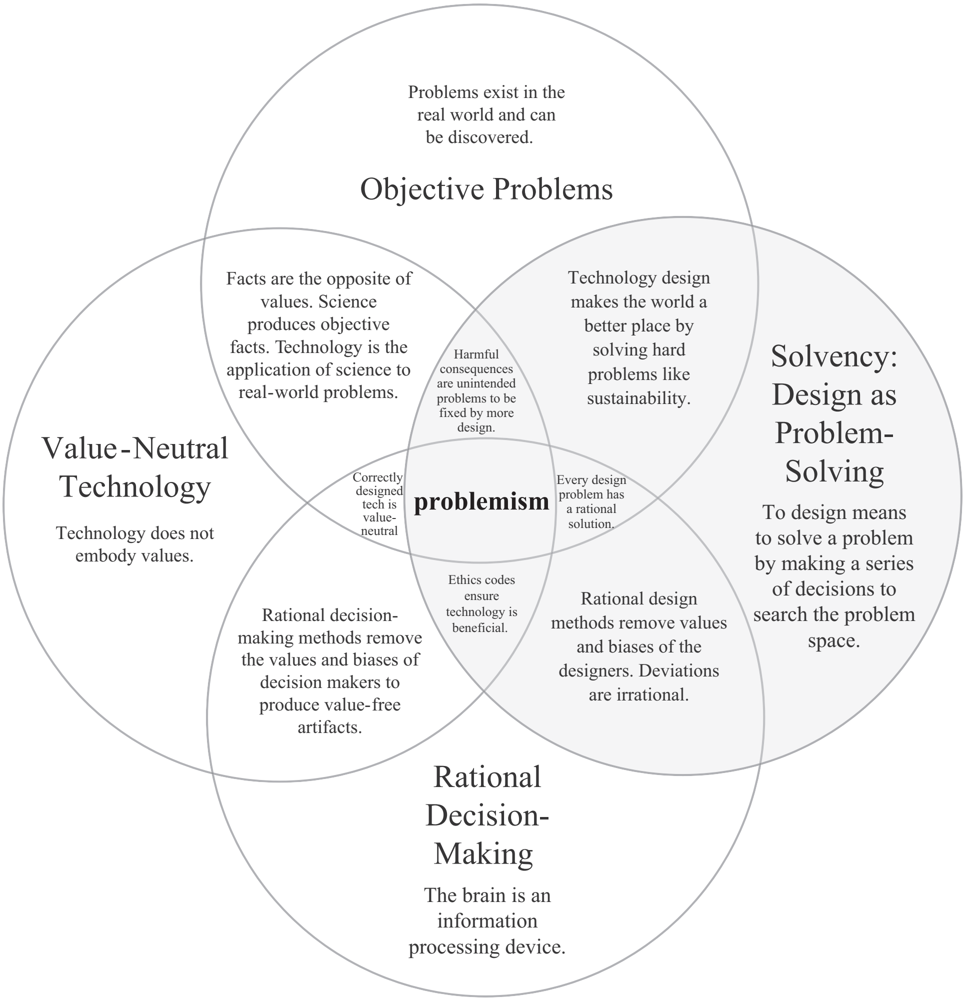
The myth of objective problems configures the boundaries of the design discourse and specifies the dimensions of the problem space. If “the problem is that people consume too much energy,” the solution must be that people need to consume less energy. Instead of expanding the view of the problem situation in which growing energy consumption is harmful to gain a systemic appreciation of its factors, the myth narrows the view by representing one particular framing as objective fact. Instead of exploring the richness of the situation through such practices as boundary critique (chapter 5) or value-sensitive design (chapter 6), the myth lends credibility to the framing presented by whoever leads the effort. It makes their knowledge claims appear as facts, and it will make marginalized voices who may speak critically about the inadequacy of one framing in the face of their lived experience appear illegitimate. For example, if the problem is framed as “consumers waste too much energy,” this implies that the problem would be solved if consumers wasted (even somewhat) less. Alternative perspectives (Knowles 2013; Strengers 2014; Brynjarsdottir et al. 2012) would highlight that: (1) Providing information about energy consumption does not necessarily lead to the desired changes in consumption. It is simply one input to situated behavior that is complex, multifaceted, and best understood in context. (2) Consumers may find themselves unable to make the desired changes because they interfere with what they perceive as their lifestyle requirements or subsistence needs. (3) While some consumers waste energy, others don’t have enough, particularly on a global scale. The richest 1 percent pollute more than twice as much as the bottom 50 percent, so changing *their* habits would have much stronger influence (Gore 2020; Chancel et al. 2021). (4) It is well established that consumer choice is not a strong leverage point for climate change in the absence of systemic change (Hickel 2020; N. Klein 2014). Corporations, specific industry sectors, and the overarching economic paradigm of growth-obsessed economic policy, all present much more effective sites for large-scale changes in energy consumption. (5) Treating people as citizens instead of consumers shifts the mindset more profoundly. In fact, it has a marked effect on their sustainability choices (Bauer et al. 2012). All this is not to say we should not foster awareness about consumer choices, but that the problem-framing prioritizes one leverage point for action and can prevent the most important conversations from taking place.

When these myths come into play all at once, they reinforce the common trope that climate change is consumers' responsibility, that technology can solve this social problem, and that we should step aside to let technologists solve it. That is the fairy tale that tech billionaires are now trying to sell.

PROBLEMISM AND THE MYTHS OF SYSTEMS DESIGN

Figure 4.2 illustrates some consequences emerging at the intersections of these myths, with problemism emerging at the center. For example, if technology is neutral (left), with neither politics nor a role reinforcing inequality, any negative consequences of design as problem-solving (right) are *unintended* and can be attributed to accidents, incomplete reasoning, or mistakes. These form new problems to be solved (top) within the existing frame, effectively displacing more nuanced perspectives. The myth of value-neutral technology thus allows problem solving to marginalize the broader consequences of its interventions.

Similarly, by framing thinking *as* information processing, the myth of rational decision-making (bottom) supports the concept of design as problem-solving performed by search (right). Rationalistic theories of choice have no mechanisms for deliberating about legitimate goals to strive for, but they provide the foundation for rationalist design methods. In fact, the original framing of bounded rationality *is* one of problem solving (Simon 1956; Simon 1996). How the frame is discursively established is out of its scope and remains invisible. Instead, the myth of objective problems (top) predicates a focus on solving those problems framed by measurable objects and entities already recognized by science over the issues voiced by those with lived experience. These problems are socially preformed, and existing technology partially preconfigures what counts as a solvable problem to begin with. The objectivity of science and its associated bureaucracy legitimate scientific expertise over other knowledges in that process. The sole focus on "correctness" and on



4.2 Problemism and the myths of computing.

the correspondence of representations to observable conditions distracts from fundamental questions of legitimacy and justification.

IS COMPUTING INSOLVENT?

Computing's role in sustainability and social justice is complicated. Despite its potential to make the world a better place, more often than not computing's harmful effects get externalized. This *debt of computing* raises the question whether computing—as a profession, an industry, and a field—will be able to pay back what it owes. This chapter advanced the position that beyond existing critiques of computing's ethical and moral positions, the very framing called problem-solving, which is so ubiquitous in computing, severely restricts *what we can talk about* when we talk about computing, human–computer interaction, and design. The myth of solvency manifests frequently, and in problemism it contributes to the rising debt of computing.

In a rigorous, values-focused discourse analysis of green computing research, Bran Knowles (2013) concludes that "the [green computing] discourse effectively commoditises 'sustainability'—making it purchasable in the form of Green ICT, persuasive apps, etc.—and in the process both 1) reifies consumerist tendencies that have driven much of the environmental destruction to date, and 2) absolves individuals from having to make more significant behaviour changes" (89). As a result, "Green Computing would need to accept major changes to the very premises upon which it is built before it is possible to a) recognise the reasons why its current strategy is unlikely to succeed, and b) craft a new strategy" (Knowles 2013, 110). Similarly, an analysis of climate-focused AI research argues that "technocentric approaches typically reduce complex human-environment relationships in ways that fail to account for social relations and power dynamics" and attests that "environmental and climate crises are grist for tech solutions" (Nost and Colven 2022, 23). In other words, this kind of work often is sustainability for IT, rather than IT for sustainability, as "many climate AI actors are interested in it for surveillance, greenwashing, and commodifying algorithms" (23) rather than to save the planet. In parallel, a review of AI ethics education attests a narrow focus that entails "disciplinary self-isolation . . . a loss of values, assumptions and methods that are crucial" (Raji, Scheuerman, and Amironesei 2021, 522). The authors conclude that this effectively renders computing graduates incapable of seeing beyond the narrow framing of technical expertise, to the detriment of real-world outcomes.

If we agree that the discourse of pure computational thought is indeed insolvent—unable to pay its debts—what should we do? In finance, insolvency can be met in two ways. The first is bankruptcy proceedings, in which assets may be liquidated to pay off outstanding debts. This would hardly be a useful approach here. Aside from the practical challenge that there are no clear entities and no judges, most of us want computing to be a part of this world—we just need it to be different. We need computing to shed what Feenberg calls the "illusion of pure rationality."

> That illusion obscures the imagination of future alternatives by granting existing technology and rationalized social arrangements an appearance of necessity they cannot legitimately claim. Critical theory demystifies this appearance to open up the future. It is neither utopian nor dystopian but situates rationality within the political where its consequences are a challenge to human responsibility. (Feenberg 2014, 167)

How should we situate the rationality of computational thought within a political concept of systems design in the face of the continued dominance of the rationalist tradition? Winograd and Flores (1986) proposed a definition of design as "the interaction between understanding and creation" (4). On this basis, they too argued that "we need to replace the rationalistic orientation if we want to understand human thought, language, and action, or to design effective computer tools" (1986, 26). Unfortunately, despite the significant influence of their book, this tradition still dominates computing culture, perhaps in part because from a perspective of technical computing, their argument remained "incomprehensible" (Agre 1997a, 39).

I will here continue the critical engagement with that tradition, invoking the other option from finance for handling insolvency: restructuring. To restructure a company's debt means to propose a detailed plan to the creditors for how the company could continue its operations while simultaneously making good on its obligations (Tuovila 2020). It seems an apt analogy for what we need.

RESTRUCTURING: WITH A LITTLE HELP FROM THE CRITICS

Stepping away from the language of problemism does not mean inaction and paralysis, but it requires more attention to care, to discourse, to deliberation. For example, some critical design approaches dissolve problemism by redesigning the system of design in which it appears. As a result, design projects built on principles of design justice (Costanza-Chock 2020) or data feminism (D'Ignazio and Klein 2020) are unlikely to exhibit severe problemism. In emerging fields such as human-centered data science in spaces with closer interactions to fields such as science and technology studies, this is already happening too (e.g., Aragon et al. 2016; Passi and Barocas 2019). But in the meantime, the rationalist tradition continues to design as well.

How can we reconcile the undisputed power of computational thinking with a renewed focus on the social and political foundations of computing practice? Not by adding the latter to the former as an afterthought, but by bringing both to the same table as equal partners. Instead of adding considerations of human factors as a separate module to our body of computing knowledge or adding one elective from the humanities or social sciences to the computer science curriculum, we need to rethink the human and social foundations of computing and design practice. When computing first became social, computer science and CS education did not fully pursue the consequences. As a result, "CS pedagogy on its own is not able to elaborate the disciplinary norms and create conditions for stable comprehensive and socially beneficent technical artifacts" (Raji, Scheuerman, and Amironesei 2021, 523). Some are already working on this task of rethinking, but it is far from completed (Connolly 2020).

CONCLUSIONS: OVERCOMING PROBLEMISM

In problemism, unreflective solutionism in the unquestioned service of dominant interests reinforces the inequitable status quo and misperceives itself as a force of enlightened progress, leaving destruction in its wake. One might say that problemism is the dominant paradigm of computing. The path to sustainability and justice begins with setting it aside.

Computing contains many pockets of critical and divergent perspectives that are hard at work repaying its debts to societies. But the mainstream narrative of computing that presents computational problem solving as the world's savior and proposes a technical rationality as the "solution" to its wicked problems is indeed unable to pay its dues. The insidious nature of problemism lies in how it lays out and prefigures the conceptual map by which we structure our view of the situations we engage in.

Instead of opening bankruptcy proceedings, I suggest we restructure computing and its understanding of systems design on a foundation that includes social and critical theory. In that suggestion, I am not alone, and many have taken crucial steps in that direction. I will not propose a grand plan or pretend to know precisely how this should happen in totality, but instead make focused and partial suggestions.

Moving beyond the myths outlined here opens a view of problem situations with ample potential for computational interventions within a critically systemic framework. By presenting problemism in a purified and somewhat sarcastic form, I hope to draw attention to the argument that *it is not enough not to be a problemist*: In systems design practice, there is no neutral ground. Instead, responsible systems design practice for the twenty-first century involves something akin to the opposite of problemism. It must be critical and systemic. Its practice must be not instrumental but critical; not deductive but dialectic; not repeatable but replicable; not seeking proof but legitimation; not aiming for optimization but justification; not universal but contingent; not rational but reasonable; not technical but sociotechnical; not complete but proudly and savvily incomplete. Developing this will be the task of the book's second part. Once we supplant computing's myths, we can build such a practice with a little, or a little more, help from the critics. There is a lot of work to do before this can become the mainstream of computing, but many have already begun. It is time for a critical turn in computing.

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