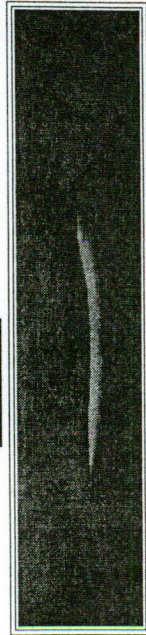




CHAIR ET MÉTAL
METAL AND FLESH



Raise high the yellow flag of quarantine:
prophylaxis against the software way of knowledge

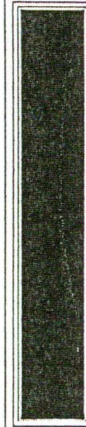
by Paulina Borsook

The author reflects upon a contemporary way of thinking linked to the pervasive place technology is taking in our individual and collective lives. Attached to a set of nested beliefs about technology is a form of epistemology - what the author calls "the software way of knowledge" - that frames and constrains the way we know and experience the world. By accepting the supremacy of software-mediated experience over any other forms of activity, we find ourselves thinking the software is "correct" and we are the ones at fault when the software makes our lives less efficient and more miserable. Software can make us experience the world in limited categories of experience, so that we may well be bypassing essential forms of "being in the world". Hence, as with any form of ideology, we tend to take for granted certain ways of dealing with our lives, ways, the author contends, are out of whack with the totality of who we really are and how the world really does work.

The author also proceeds to review thinkers from the present and recent past who inspired her thoughts on technological determinism and presents their views on related topics: theories of mind, effects of the use of technology on the individual and on social structures, and job-deskilling, technology mediated education, etc. A common ground to these reflections is found in the spectre of an alienated society of dehumanized citizens.

book by Paulina Borsook

[contact](#)



Raise high the yellow flag of quarantine: prophylaxis against the software way of knowledge

By Paulina Borsook

I've knocked around high-tech since the early 1980s, because even back then, in the Bay Area, for liberal-arts flakes, that's where the jobs were. It was only after I had been immersed in high-tech for several years that I began to notice, well, something. I began to call the something I was seeing 'that', because I didn't know exactly what it was, but I knew I kept on seeing it. I couldn't have described the 'that' that well to anyone, and at best the 'that' was the vaguest of recurrent preverbal constructs, an 'ah, there's -that- again...' Anyway, it took me years to figure out that the 'that' I was seeing were instantiations of the dominant libertarian culture of high-tech; and it took me several more years, even up to the point when I was helping my editor write the jacket blurbs for "Cyberselfish", my book all about the 'that', that the 'that' had not been the politics or philosophy of high-tech, but its -religion-, if religion is understood to be a mostly unconscious, commonly held, collective beliefs.

So I want to talk about a similar kind of "that" today, only my hunch is that this time what I am glimpsing from the corner of my eyes repeatedly is not religion but epistemology; that is, it's something I've taken to calling the software way of knowledge. It's a 'that' that I think is leaking out from high-tech to the world at large as alas, it should not. There's a fancy academic-theorist term for keeping what belongs to technology within the realm of technology: it's "technospecificity". But "technospecificity" doesn't help us figure out the -what- from the world of high-tech that is contaminating the aquifers that support the rest of our lives.

It's a little nervous-making to try to define the software way of knowledge, for not only is it a bit like trying to define "weather" (that is, something which surrounds us, we take for granted, and notice more or less from time to time). It's also scary to try to describe something which is sensed rather than known, that's so ambient and pervasive that it's not even as concrete as other, seemingly similar atmospheric ideas such as the supposedly left-wing bias of the media, or the anxiety of influence Hemingway is purported to hold over all American writers who've followed him.

So to be alarmingly reductive, what I think of as the software way of knowledge consists of certain kinds of truisms: that applying software to our lives inevitably makes them better; and more software means more betterness. That the ways of thinking software demands --- discrete units of data, pre-determined transactions, no ambiguity and no ellipses --- is better, no matter how much we ache in the effort to contort ourselves to software's exigencies. And as we are increasingly slaved to software-mediated experiences in our lives, we begin to accept that it can't be any other way.

Which is NOT to say I am on the side of Sven Birkerts' "Gutenberg Elegies" or anything else that says computer hardware and software are all for the bad, and signify endtimes even more than the rise of television studies in universities, or the catastrophic transformations we have all been going through since 9/11. I think computers are just ducky in their place --- I could never have become a writer without them, because I cannot type, and suffer from terrific writer's block, so without being able to backspace in light, I would have been as disabled as say, if I had been born with poor eyesight in the long centuries before the creation of spectacles. But I do have a perception of increased social and personal toxicity from software --- and that's the point of trying to discuss the software way of knowledge.

I have a friend who in her junior year switched from majoring in art to making the choices that led her to acquiring her masters in software engineering. This is a woman who worked in the field of artificial intelligence for many years, was a grand high poohbah at one of those feared Big Five technical consultancies, and has even been a CEO of a -legit- high-tech startup. When I asked for her take on this second, elusive, high-tech 'that', she was a bit like a pica musing in

her alpine fastness, puzzled but alert. It took her a bit, but she finally said, "Oh -that-!" (yes, another instance of "that") and recounted one of her own experience as a manager. As part of the responsibilities involved with her running the high-tech skunkworks of an internationally famous toy company (no names, please), she wanted to improve efficiency by using software to centralize the answering of phones. But her admins were smarter than she was; they pointed out to her that it was important for visitors to see someone who might know what was going on when they walked in, and it was important for callers from outside the division to have a person, and not a bot, to talk to.

What they were asking for, and which my wise friend eventually figured out was the correct course of action, was for the actually greater, but human-touched, efficiency in the workplace. Contrast this to our familiar experience of being lost in the voicemail hell which we all know is not -really- more efficient in terms of accomplishing anything, but which simply offloads the work, and the costs of the time, of the transaction onto the caller, and because of its oversimplification of choices, eliminates the possibility of many interactions.

Michael Dertouzos, who had been director of MIT's Laboratory of Computer Science for years until his death a few months ago, published a book in late 2000 called "The Unfinished Revolution/Making computers human-centric", ostensibly a computer scientist's attempt to move out beyond the limitations computers have placed on the humans that have to work with them. But "The Unfinished Revolution" is as fine a proof as one could ask for of the tenaciousness of the software way of knowledge: in spite of his somewhat accurate diagnosis of the failures of modern software to deliver on its promise of more ease and less aggravation, Dertouzos defaults to yet another of the postulates of the epistemology: that the bottlenecks in our lives come from not getting the right data fast enough. It's not the horrors of managed care, or the shoddy salaries paid teachers, that are the problem: it's not being able to synch your Palm to the remotely-imaged X-ray, or not being able to maintain adequate per-capita allotment of laptops in predominantly poor school districts. It's so much more fashionable and fun to get indignant about the digital divide than to worry about the decline in real wages, the impossibility of living on minimum wage, and the loss of blue-collar jobs to overseas factories with sweatshop conditions --- we are, after all, one nation under the god of global capitalism, you betcha, and if we all worked with computers the future would be so bright we'd have to wear shades. Never mind the thousands of high-tech workers now out of jobs, or the billions of dollars in unsold inventory of computer and communications equipment now threatening to contaminate our landfills with lead and worse.

As an aside, it's worth mentioning that the glut of computers and communications equipment is now great enough that manufacturers such as Cisco will -buy- back at bankruptcy auction the barely-used gear dead dotcoms such as Webvan purchased --- just to take it off the market with the hope of somewhat shoring up the purchase price of new gear. In a sense, the whole dotcom new economy irrational exuberance can be seen as the software way of knowledge gone berzerk: that somehow, high-tech itself, and the fair-weather fans it attracted in the mid to late 90s, forgot that computer software and hardware are supposed to be in -support- of something else (a company, a non-profit), are a means to an end, and are not an end in themselves. But the high-tech's delightful adolescent narcissism somehow took hold in the rest of the land; and its peculiar ways of slicing and dicing the world were held, for a time, as self-evident truths applicable across all lines of human striving and economic behavior. We know one of the causes behind the dotgone dotbomb techwreck was that the software way of knowledge trumped common sense: that you potentially -could- do something online, didn't mean it was -better- done online.

Suffice to say, the stubbornness of this epistemological framework is revealed in the way our pal Dertouzos, and pretty much every other computer scientist tasked with thinking about new technology, keeps bellyfloping onto the same stale yesterday's tomorrows that John Sculley first popularized in the "Knowledge Navigator" video back in the 1980s, the technoutopia of intelligent agents and everything being connected to everything else all the time. Persistence of vision, indeed --- such wacky astigmatisms, if quarantined to the small group of people afflicted with them, might not matter. But the distorted products and worldviews they generate bump into us everywhere --- and that's not pleasant. As much as I love El Greco, (and no current

commercial designer of software could rank with him, though I might consider some computer pioneers, such as Paul Baran or Claude Shannon or John Von Neumann or Alan Turing or Doug Engelbart, in his class), I wouldn't want to live in a house designed for the figures he paints, or try to wear the clothes that might fit them.

The weird thing is, though, the software way of knowledge has been known --- and debunked --- for going on thirty years. It's worth mentioning some of its most nuanced --- and hence, best --- debunkers. First and foremost there's Hubert Dreyfus, a philosophy professor at Berkeley, who wrote a book called "What computers can't do" which was published back in 1972, followed on by the book he wrote with his brother, a professor of industrial engineering also at Berkeley, entitled "Mind over Machine/The power of human intuition and expertise in the era of the computer", published in 1986. An updated version of the first book, "What computers -still- can't do", was published in 1994. Initially engendered as critiques of the attempts on artificial intelligence, the arguments in these books can be deployed as excellent elevator-pitches/soundbytes/cribsheets to arm yourself against all hypothetical arguments that suggest something like that we can understand the mind from how computers work, and that computers will ultimately be able to have consciousness, as we as ordinary mortals, and not as possible aficionados of science fiction, understand the term. That the brain is like a computer, or a computer can be like a brain, when the root of our genius and our individuality lies in our sensorium and our experience and our emotional associations --- that is, our subjectivity harnessed to our logical capacities --- are two of the most long-lived and pernicious bits of nonsense generated by the software way of knowledge. Yes, take that, Stephen Pinker.

The Dreyfus brothers systematically destroy, as only a philosopher and an engineer can do, what they call "the assumptions underlying the persistent optimism" that have been continuously asserted from the 1950s to today: that is, - the biological assumption that the brain operates in some sort of equivalent of on/off switches - the psychological assumption that the mind operates on bits of information according to formal rules - the epistemological assumption that all knowledge can be formalized in terms of logical relations - the ontological assumption that the sets of facts that describe the world are context-free determinate elements.

Or as I would put it with less intellectual rigor, we are not just processors of texts or numbers --- but these processes lend themselves to the software way of knowledge as associative and divergent thinking do not. We are more than meatspace Microsoft Word and Excel. Regardless, I recommend these books to you, and won't waste your time recapitulating all the points the Brothers Dreyfus so excellently made. Another guy who's had long-standing issues with all this is sociologist Theodore Rozak. A decade before he published his 1994 "Cult of Information", more or less a thorough and reasoned critique of the software way of knowledge, he succinctly described the profound failure of educational software, and computers in education, in "Processed World", an anarcho-situationist zine which started in San Francisco in the early 1980s, and was the first publication to take -data processing- and the life it created, as its main subject. Be that as it may, at the very dawning of the era when parents began to get flippy over whether their children would be computer-literate, if such a thing can be said to exist (seems to me if you can read, write, and reason, you can ultimately figure out how to work with what are ultimately dumb machines), Rozak was in there with facts, figures, and anecdotes, wailing on the imagination-stunting, intellectual-movement-limiting, visual-vocubularly crimping effects of the pathways computers force kids into.

Most people learn best when there is an emotional involvement with ideas and their presenters. What's more, a dirty little secret about class divisions in contemporary society is that an online degree isn't valued as much as one acquired in residence: so the lumpen who have to work fulltime get the low-status online credential from the University of Phoenix; the elite kids get to actually attend school, and have the rich random intersections, and making of connections, that most folks say are what they valued from their educations. That, and the encounter with a wondrous teacher, a good lab, a great library. None of which virtuality can provide --- or purports to value.

It's not entirely surprising that even within high-tech itself, though, there are understandings about the problems with the software way of knowledge; even though high-tech tends to filter

for the IQ-smart, rather than the reflective, those kinds of personalities still make their appearance in that industry sector, as they turn up in any other.

Alan Cooper, a pioneer of microcomputer programming, published in 1999 "The inmates are running the asylum/Why high-tech products drive us crazy and how to restore sanity", about the enraging clunkiness of software and interface design. His book detailed the ways that every day, computers are making people easier to use --- and how people feel ashamed when they can't conform to the machine, that is, are not power users. The cost of not being literally perfect (mind you, I didn't say logical: as anyone who has actually tried to use software --- or worse, create software, can tell you --- software can be as inconsistent and confounding and erratic as what any 19th-century woman was reputed to be) is so great that it encourages people to become detail-oriented pedants.

Whereas synthesis, creativity, and spirituality --- just about the best stuff we humans can do or experience --- rely on other means of getting there, a truth some of the best scientists (even computer scientists such as Carver Mead and Donald Knuth) will admit to, privately.

So much of what's ineffable turns out to be what really matters. Richard Sennett is an eloquent sociologist who years ago wrote the classic "Hidden injuries of class", about the unconscious psychological limitations and entitlements fostered by supposedly non-existent class structure in the United States. His recent "The corrosion of character/The personal consequences of work in the new capitalism", among other things, describes the difference in the psychological worklives of folks working in a bakery where 30 years ago, there was a strong union, camaraderie, and esprit de corps, regardless of other kinds of self-limiting damage that derived from the traditional, class-bound ways they saw themselves. Now, software-based efficiency has come to the bakery. How this plays out is that baking is done programmatically by machine; when the machines break down, it's something with the software and a geek needs to be called; no one who works there now actually knows how to bake or has the pleasure of baking; and the workforce is all part-time, and no one knows each other.

Sennett's book is in somewhat of the same spirit of Robert Putnam's "Bowling Alone", about the decline of social capital and the rise of end-of-the-day/net-net/bottom-line kinds of thinking throughout modern life; but software also operates as a covert agent of social decay and anomie throughout the free-agent Hobbesian world Sennett limns and which sadly we have come to expect.

I have wondered if a tiny, unremarked aspect of what people found so moving about the WTC tragedy is relevant to our unease with the software way of knowledge. The loving repetition of the tales of the deep human bonds of comradeship amongst the firemen and police officers, the pity and terror invoked by those bonds sundered by death in heroic rescue attempts, the depth of folks' response to these narratives --- these may be in small part a recognition of our need for social capital, which both Sennett and Putnam outline. And in a way, signify a cultural rebellion against the software way of knowledge. That is, those endlessly reported-on stories repeated how these guys had lived and worked together; and their valor somehow was something of an extension of the ties of human connection that they had established with their buddies. These stories were a shock to our system, served as a reminder of how much we crave the permanence of real-life/real-time human connection --- something not replaceable by that found on IRC.

We pine for the hands-on. In fact, neurologist Frank Wilson's wonderful book "The hand" is, as its subtitle states, "how its use shapes the brain, language, and human culture". But what it's really about is the genius of our hand-eye coordination, our understandings derived from manual dexterity. He tells of the experience of one of the Big Three automakers, which used to make a habit of hiring kids straight out of MIT with the relevant engineering degrees. But the company determined after a time that this was not good screening device for new hires in and of itself, for the kids who had never actually -worked- on cars but had had plenty of experience with software, particularly of the design kind, weren't capable of truly thinking three-dimensionally, coming up with any innovative, or really knew in their bones what carness meant. Similarly, architects age 40 or older complain that young architects can't draw and can't really think about design outside of the templates that the software they are habituated to imprisons them with.

Wilson's technical term for this somatic, experiential knowing is felt paths --- and is not so very different from an elemental concept the Dreyfus' brothers delineated --- the difference between knowing -how- (that is, how to ride a bike) and knowing -that- (trigonometric formulae). It's the difference between intuitive expertise (with understandings humans arrive at over time based on, well, experience) and expert systems (with facts and rules that can be articulated, which you might say are programmable). And empirical evidence shows for the most part that intuitive expertise trumps expert systems, most of the time.

For we do live in the world of experience, and not in a data warehouse and not in the movie "The Matrix". Years ago, I attended one of the annual CSCW conferences, CSCW standing for computer-supported collaborative work, a kind special-interest research group on human-factors in computing. The conference itself is sponsored by the Association for Computing Machinery, sort of the AMA of computer science. Anyway, among the people giving papers was an esteemed scientist from Bell Labs, back when Bell Labs was still a world-heritage scientific resource, and hadn't been dissolved away in pursuit of the stockholder theory of value. In his talk, the scientist confessed that the way he actually began research on a new subject that he didn't know much about, didn't know the literature of, didn't know exactly what he was wanting to discover about --- was to go to the library and stare at the shelves of books in the subject area, and see what struck him. It was intuition, and human pattern-recognition, and not keywords, that worked for him.

Nicholson Baker would have agreed with this Bell Labs fellow, for he has written extensively about the knowledge-losses involved in putting book catalogs online, how online information is no substitute for books or libraries or librarians, how newspapers need to remain as print and not as microfiche, and the deficiencies of reliance on keywords --- and I commend his writings on these subjects to you, and don't want to do so much more poorly than what he has done well.

But I do want to contrast the insight of that Bell Labs scientist with the puzzlement of a young man working in the main library on the UC-Berkeley campus. I hadn't been to Doe Library in a few years, and so wasn't sure where things had been moved to. So I asked him 'where is the catalog?' He looked puzzled, incomprehending. So I tried again. "Ok, I know the catalog has been put online, but where do I get access to it?" Again, no understanding of what this thing called a 'catalog' was I was asking for. Finally, he said "there is computer access across the hall, and it has all the information resources on it". He no longer had an understanding that there was a thing called a catalog; that there might be such a value in such a thing. Or, that a -database-, by extension, would of necessity be more inclined to errors and omissions than a discrete bounded body of knowledge such as a -catalog-. And yes, there were annoying errors in the online listings of books that a card catalog wouldn't have contained.

And the thing is, we all secretly know the flaws in the software way of knowledge. You increasingly have to pay to talk to a -human- who can help you with software support --- but we know it's only a human who can help us through whatever illogical counterintuitive non-productive rat-hole the software has gotten us into.

Some of you may be familiar with the dotcom deadpool/expose website www.fuckedcompany.com, just about the best and most wicked online resource I know for understanding what went wrong in the way new kewl economy. Anyway, a friend of Philip Kaplan, the evil young genius who founded the website, has recently launched a promotional campaign that markets his high-end male underwear line by making the pitch "you listen to high-end digital music, you insist on state of the art computers". But the fact is folks at the high-end can opt -out- of the software way of knowledge. They can buy artisan bread made solely by humans and without software; they have the option of being untethered from pagers and cell-phones and email, having personal assistants who are human (a.k.a house slaves) doing the sorting and the managing of information and access, and not depending so predominantly on filters and pre-set software parameters.

Conversely service workers, those so-necessary and so-maltreated girders of our economy, are always the victims of the software way of knowledge, subjected without choice to the latest rev of whatever --- which endlessly complicates their worklives, which is more brittle and more breakable than what they worked with before, and which takes no account of their native human

gift and need for flexibility, as opposed to what might fit into the fields in a form. Being forced into the hierarchical thinking of decision-trees --- when life is -analog- and not digital is painful and impoverishing. But somehow we've been seduced into thinking that we are not feeling what we are feeling --- or that we shouldn't be feeling it at all.

We know the ways resume-scanning programs so often don't map to our actual work experience. We know how frustrating to the point of tears it can be to try to correct an error in computerized billing. We know how fruitless it can actually be to go mortgage-shopping, for most of the brokers are using the same software performing the same calculations --- so you really don't have the option of dealing with a personal banker who really gets, and can make an exception for, your unique circumstances.

The current romance with sociobiology, particularly popular among geeks, is an example of the pervasive infiltration of the idea that everything can be reduced to that which could be mapped out through an auction program. That is, reducing behavior, particularly the mating dance, to variables that can be quantified such as economic resources and measurable social status, and happy heuristics such as eggs are scarce and sperm is cheap, are comforting in its pseudo-mathematical simplicity. But doesn't explain why the girls at the party are far more interested in the indie film director than the engineer who at least until recently, made four times the salary and had four times the job security of the guy who could make interesting conversation.

In our post-Y2K, post-apocalyptic world, the software way of knowledge has come to stand in for the demonic bureaucracy Kafka first explained to us in "The Trial", "The Castle" and "In the Penal Colony": it's dehumanizing, it doesn't make any sense, it doesn't fit who we really are, but somehow we come to believe the fault lies within us.

So the same malefic force that contributes to the travails of a Nordstrom saleswoman or a travel-agent partially explains why so much of modern architecture is boring or a failure. It's sort of relief to recognize that some of what ails us is that, which up til the NASDAQ crash, was so celebrated. It's the software, stupid.