

you
are
not
a
gadget

A MANIFESTO
JARON
LANIER

ALFRED A. KNOPF
NEW YORK
2010

MISSING PERSONS

SOFTWARE EXPRESSES IDEAS about everything from the nature of a musical note to the nature of personhood. Software is also subject to an exceptionally rigid process of "lock-in." Therefore, ideas (in the present era, when human affairs are increasingly software driven) have become more subject to lock-in than in previous eras. Most of the ideas that have been locked in so far are not so bad, but some of the so-called web 2.0 ideas are stinkers, so we ought to reject them while we still can.

Speech is the mirror of the soul; as a man speaks, so is he.

PUBLILIUS SYRUS

Fragments Are Not People

Something started to go wrong with the digital revolution around the turn of the twenty-first century. The World Wide Web was flooded by a torrent of petty designs sometimes called web 2.0. This ideology promotes radical freedom on the surface of the web, but that freedom, ironically, is more for machines than people. Nevertheless, it is sometimes referred to as "open culture."

Anonymous blog comments, vapid video pranks, and lightweight mashups may seem trivial and harmless, but as a whole, this widespread practice of fragmentary, impersonal communication has demeaned interpersonal interaction.

Communication is now often experienced as a superhuman phenomenon that towers above individuals. A new generation has come of age with a reduced expectation of what a person can be, and of who each person might become.

The Most Important Thing About a Technology Is How It Changes People

When I work with experimental digital gadgets, like new variations on virtual reality, in a lab environment, I am always reminded of how small changes in the details of a digital design can have profound unforeseen effects on the experiences of the humans who are playing with it. The slightest change in something as seemingly trivial as the ease of use of a button can sometimes completely alter behavior patterns.

For instance, Stanford University researcher Jeremy Bailenson has demonstrated that changing the height of one's avatar in immersive virtual reality transforms self-esteem and social self-perception. Technologies are extensions of ourselves, and, like the avatars in Jeremy's lab, our identities can be shifted by the quirks of gadgets. It is impossible to work with information technology without also engaging in social engineering.

One might ask, "If I am blogging, twittering, and wikiing a lot, how does that change who I am?" or "If the 'hive mind' is my audience, who am I?" We inventors of digital technologies are like stand-up comedians or neurosurgeons, in that our work resonates with deep philosophical questions; unfortunately, we've proven to be poor philosophers lately.

When developers of digital technologies design a program that requires you to interact with a computer as if it were a person, they ask you to accept in some corner of your brain that you might also be conceived of as a program. When they design an internet service that is edited by a vast anonymous crowd, they are suggesting that a random crowd of humans is an organism with a legitimate point of view.

Different media designs stimulate different potentials in human nature. We shouldn't seek to make the pack mentality as efficient as possible. We should instead seek to inspire the phenomenon of individual intelligence.

"What is a person?" If I knew the answer to that, I might be able to program an artificial person in a computer. But I can't. Being a person is not a pat formula, but a quest, a mystery, a leap of faith.

Optimism

It would be hard for anyone, let alone a technologist, to get up in the morning without the faith that the future can be better than the past.

Back in the 1980s, when the internet was only available to small number of pioneers, I was often confronted by people who feared that the strange technologies I was working on, like virtual reality, might unleash the demons of human nature. For instance, would people become addicted to virtual reality as if it were a drug? Would they become trapped in it, unable to escape back to the physical world where the rest of us live? Some of the questions were silly, and others were prescient.

How Politics Influences Information Technology

I was part of a merry band of idealists back then. If you had dropped in on, say, me and John Perry Barlow, who would become a cofounder of the Electronic Frontier Foundation, or Kevin Kelly, who would become the founding editor of *Wired* magazine, for lunch in the 1980s, these are the sorts of ideas we were bouncing around and arguing about. Ideals are important in the world of technology, but the mechanism by which ideals influence events is different than in other spheres of life. Technologists don't use persuasion to influence you—or, at least, we don't do it very well. There are a few master communicators among us (like Steve Jobs), but for the most part we aren't particularly seductive.

We make up extensions to your being, like remote eyes and ears (webcams and mobile phones) and expanded memory (the world of details you can search for online). These become the structures by which you

connect to the world and other people. These structures in turn can change how you conceive of yourself and the world. We tinker with your philosophy by direct manipulation of your cognitive experience, not indirectly, through argument. It takes only a tiny group of engineers to create technology that can shape the entire future of human experience with incredible speed. Therefore, crucial arguments about the human relationship with technology should take place between developers and users before such direct manipulations are designed. This book is about those arguments.

The design of the web as it appears today was not inevitable. In the early 1990s, there were perhaps dozens of credible efforts to come up with a design for presenting networked digital information in a way that would attract more popular use. Companies like General Magic and Xanadu developed alternative designs with fundamentally different qualities that never got out the door.

A single person, Tim Berners-Lee, came to invent the particular design of today's web. The web as it was introduced was minimalist, in that it assumed just about as little as possible about what a web page would be like. It was also open, in that no page was preferred by the architecture over another, and all pages were accessible to all. It also emphasized responsibility, because only the owner of a website was able to make sure that their site was available to be visited.

Berners-Lee's initial motivation was to serve a community of physicists, not the whole world. Even so, the atmosphere in which the design of the web was embraced by early adopters was influenced by idealistic discussions. In the period before the web was born, the ideas in play were radically optimistic and gained traction in the community, and then in the world at large.

Since we make up so much from scratch when we build information technologies, how do we think about which ones are best? With the kind of radical freedom we find in digital systems comes a disorienting moral challenge. We make it all up—so what shall we make up? Alas, that dilemma—of having so much freedom—is chimerical.

As a program grows in size and complexity, the software can become a cruel maze. When other programmers get involved, it can feel like a labyrinth. If you are clever enough, you can write any small program from scratch, but it takes a huge amount of effort (and more than a little

luck) to successfully modify a large program, especially if other programs are already depending on it. Even the best software development groups periodically find themselves caught in a swarm of bugs and design conundrums.

Little programs are delightful to write in isolation, but the process of maintaining large-scale software is always miserable. Because of this, digital technology tempts the programmer's psyche into a kind of schizophrenia. There is constant confusion between real and ideal computers. Technologists wish every program behaved like a brand-new, playful little program, and will use any available psychological strategy to avoid thinking about computers realistically.

The brittle character of maturing computer programs can cause digital designs to get frozen into place by a process known as lock-in. This happens when many software programs are designed to work with an existing one. The process of significantly changing software in a situation in which a lot of other software is dependent on it is the hardest thing to do. So it almost never happens.

Occasionally, a Digital Eden Appears

One day in the early 1980s, a music synthesizer designer named Dave Smith casually made up a way to represent musical notes. It was called MIDI. His approach conceived of music from a keyboard player's point of view. MIDI was made of digital patterns that represented keyboard events like "key-down" and "key-up."

That meant it could not describe the curvy, transient expressions a singer or a saxophone player can produce. It could only describe the tile mosaic world of the keyboardist, not the watercolor world of the violin. But there was no reason for MIDI to be concerned with the whole of musical expression, since Dave only wanted to connect some synthesizers together so that he could have a larger palette of sounds while playing a single keyboard.

In spite of its limitations, MIDI became the standard scheme to represent music in software. Music programs and synthesizers were designed to work with it, and it quickly proved impractical to change or dispose of all that software and hardware. MIDI became entrenched, and despite Herculean efforts to reform it on many occasions by a multi-

decade-long parade of powerful international commercial, academic, and professional organizations, it remains so.

Standards and their inevitable lack of prescience posed a nuisance before computers, of course. Railroad gauges—the dimensions of the tracks—are one example. The London Tube was designed with narrow tracks and matching tunnels that, on several of the lines, cannot accommodate air-conditioning, because there is no room to ventilate the hot air from the trains. Thus, tens of thousands of modern-day residents in one of the world's richest cities must suffer a stifling commute because of an inflexible design decision made more than one hundred years ago.

But software is worse than railroads, because it must always adhere with absolute perfection to a boundlessly particular, arbitrary, tangled, intractable messiness. The engineering requirements are so stringent and perverse that adapting to shifting standards can be an endless struggle. So while lock-in may be a gangster in the world of railroads, it is an absolute tyrant in the digital world.

Life on the Curved Surface of Moore's Law

The fateful, unnerving aspect of information technology is that a particular design will occasionally happen to fill a niche and, once implemented, turn out to be unalterable. It becomes a permanent fixture from then on, even though a better design might just as well have taken its place before the moment of entrenchment. A mere annoyance then explodes into a cataclysmic challenge because the raw power of computers grows exponentially. In the world of computers, this is known as Moore's law.

Computers have gotten *millions* of times more powerful, and immensely more common and more connected, since my career began—which was not so very long ago. It's as if you kneel to plant a seed of a tree and it grows so fast that it swallows your whole village before you can even rise to your feet.

So software presents what often feels like an unfair level of responsibility to technologists. Because computers are growing more powerful at an exponential rate, the designers and programmers of technology must be extremely careful when they make design choices. The consequences

of tiny, initially inconsequential decisions often are amplified to become defining, unchangeable rules of our lives.

MIDI now exists in your phone and in billions of other devices. It is the lattice on which almost all the popular music you hear is built. Much of the sound around us—the ambient music and audio beeps, the ring-tones and alarms—are conceived in MIDI. The whole of the human auditory experience has become filled with discrete notes that fit in a grid.

Someday a digital design for describing speech, allowing computers to sound better than they do now when they speak to us, will get locked in. That design might then be adapted to music, and perhaps a more fluid and expressive sort of digital music will be developed. But even if that happens, a thousand years from now, when a descendant of ours is traveling at relativistic speeds to explore a new star system, she will probably be annoyed by some awful beepy MIDI-driven music to alert her that the antimatter filter needs to be recalibrated.

Lock-in Turns Thoughts into Facts

Before MIDI, a musical note was a bottomless idea that transcended absolute definition. It was a way for a musician to think, or a way to teach and document music. It was a mental tool distinguishable from the music itself. Different people could make transcriptions of the same musical recording, for instance, and come up with slightly different scores.

After MIDI, a musical note was no longer just an idea, but a rigid, mandatory structure you couldn't avoid in the aspects of life that had gone digital. The process of lock-in is like a wave gradually washing over the rulebook of life, culling the ambiguities of flexible thoughts as more and more thought structures are solidified into effectively permanent reality.

We can compare lock-in to scientific method. The philosopher Karl Popper was correct when he claimed that science is a process that disqualifies thoughts as it proceeds—one can, for example, no longer reasonably believe in a flat Earth that sprang into being some thousands of years ago. Science removes ideas from play empirically, for good reason.

Lock-in, however, removes design options based on what is easiest to program, what is politically feasible, what is fashionable, or what is created by chance.

Lock-in removes ideas that do not fit into the winning digital representation scheme, but it also reduces or narrows the ideas it immortalizes, by cutting away the unfathomable penumbra of meaning that distinguishes a word in natural language from a command in a computer program.

The criteria that guide science might be more admirable than those that guide lock-in, but unless we come up with an entirely different way to make software, further lock-ins are guaranteed. Scientific progress, by contrast, always requires determination and can stall because of politics or lack of funding or curiosity. An interesting challenge presents itself: How can a musician cherish the broader, less-defined concept of a note that preceded MIDI, while using MIDI all day long and interacting with other musicians through the filter of MIDI? Is it even worth trying? Should a digital artist just give in to lock-in and accept the infinitely explicit, finite idea of a MIDI note?

If it's important to find the edge of mystery, to ponder the things that can't quite be defined—or rendered into a digital standard—then we will have to perpetually seek out entirely new ideas and objects, abandoning old ones like musical notes. Throughout this book, I'll explore whether people are becoming like MIDI notes—overly defined, and restricted in practice to what can be represented in a computer. This has enormous implications: we can conceivably abandon musical notes, but we can't abandon ourselves.

When Dave made MIDI, I was thrilled. Some friends of mine from the original Macintosh team quickly built a hardware interface so a Mac could use MIDI to control a synthesizer, and I worked up a quick music creation program. We felt so free—but we should have been more thoughtful.

By now, MIDI has become too hard to change, so the culture has changed to make it seem fuller than it was initially intended to be. We have narrowed what we expect from the most commonplace forms of musical sound in order to make the technology adequate. It wasn't Dave's fault. How could he have known?

Digital Reification: Lock-in Turns Philosophy into Reality

A lot of the locked-in ideas about how software is put together come from an old operating system called UNIX. It has some characteristics that are related to MIDI.

While MIDI squeezes musical expression through a limiting model of the actions of keys on a musical keyboard, UNIX does the same for all computation, but using the actions of keys on typewriter-like keyboards. A UNIX program is often similar to a simulation of a person typing quickly.

There's a core design feature in UNIX called a "command line interface." In this system, you type instructions, you hit "return," and the instructions are carried out.* A unifying design principle of UNIX is that a program can't tell if a person hit return or a program did so. Since real people are slower than simulated people at operating keyboards, the importance of precise timing is suppressed by this particular idea. As a result, UNIX is based on discrete events that don't have to happen at a precise moment in time. The human organism, meanwhile, is based on continuous sensory, cognitive, and motor processes that have to be synchronized precisely in time. (MIDI falls somewhere in between the concept of time embodied in UNIX and in the human body, being based on discrete events that happen at particular times.)

UNIX expresses too large a belief in discrete abstract symbols and not enough of a belief in temporal, continuous, nonabstract reality; it is more like a typewriter than a dance partner. (Perhaps typewriters or word processors ought to always be instantly responsive, like a dance partner—but that is not yet the case.) UNIX tends to "want" to connect to reality as if reality were a network of fast typists.

If you hope for computers to be designed to serve embodied people as well as possible people, UNIX would have to be considered a bad design. I discovered this in the 1970s, when I tried to make responsive musical

*The style of UNIX commands has, incredibly, become part of pop culture. For instance, the URLs (universal resource locators) that we use to find web pages these days, like <http://www.jaronlanier.com/>, are examples of the kind of key press sequences that are ubiquitous in UNIX.

instruments with it. I was trying to do what MIDI does not, which is work with fluid, hard-to-notate aspects of music, and discovered that the underlying philosophy of UNIX was too brittle and clumsy for that.

The arguments in favor of UNIX focused on how computers would get literally millions of times faster in the coming decades. The thinking was that the speed increase would overwhelm the timing problems I was worried about. Indeed, today's computers are millions of times faster, and UNIX has become an ambient part of life. There are some reasonably expressive tools that have UNIX in them, so the speed increase has sufficed to compensate for UNIX's problems in some cases. But not all.

I have an iPhone in my pocket, and sure enough, the thing has what is essentially UNIX in it. An unnerving element of this gadget is that it is haunted by a weird set of unpredictable user interface delays. One's mind waits for the response to the press of a virtual button, but it doesn't come for a while. An odd tension builds during that moment, and easy intuition is replaced by nervousness. It is the ghost of UNIX, still refusing to accommodate the rhythms of my body and my mind, after all these years.

I'm not picking in particular on the iPhone (which I'll praise in another context later on). I could just as easily have chosen any contemporary personal computer. Windows isn't UNIX, but it does share UNIX's idea that a symbol is more important than the flow of time and the underlying continuity of experience.

The grudging relationship between UNIX and the temporal world in which the human body moves and the human mind thinks is a disappointing example of lock-in, but not a disastrous one. Maybe it will even help make it easier for people to appreciate the old-fashioned physical world, as virtual reality gets better. If so, it will have turned out to be a blessing in disguise.

Entrenched Software Philosophies Become Invisible Through Ubiquity

An even deeper locked-in idea is the notion of the file. Once upon a time, not too long ago, plenty of computer scientists thought the idea of the file was not so great.

The first design for something like the World Wide Web, Ted Nelson's Xanadu, conceived of one giant, global file, for instance. The first iteration of the Macintosh, which never shipped, didn't have files. Instead, the whole of a user's productivity accumulated in one big structure, sort of like a singular personal web page. Steve Jobs took the Mac project over from the fellow who started it, the late Jef Raskin, and soon files appeared.

UNIX had files; the Mac as it shipped had files; Windows had files. Files are now part of life; we teach the idea of a file to computer science students as if it were part of nature. In fact, our conception of files may be more persistent than our ideas about nature. I can imagine that someday physicists might tell us that it is time to stop believing in photons, because they have discovered a better way to think about light—but the file will likely live on.

The file is a set of philosophical ideas made into eternal flesh. The ideas expressed by the file include the notion that human expression comes in severable chunks that can be organized as leaves on an abstract tree—and that the chunks have versions and need to be matched to compatible applications.

What do files mean to the future of human expression? This is a harder question to answer than the question "How does the English language influence the thoughts of native English speakers?" At least you can compare English speakers to Chinese speakers, but files are universal. The idea of the file has become so big that we are unable to conceive of a frame large enough to fit around it in order to assess it empirically.

What Happened to Trains, Files, and Musical Notes Could Happen Soon to the Definition of a Human Being

It's worth trying to notice when philosophies are congealing into locked-in software. For instance, is pervasive anonymity or pseudonymity a good thing? It's an important question, because the corresponding philosophies of how humans can express meaning have been so ingrained into the interlocked software designs of the internet that we might never be able to fully get rid of them, or even remember that things could have been different.

We ought to at least try to avoid this particularly tricky example of impending lock-in. Lock-in makes us forget the lost freedoms we had in the digital past. That can make it harder to see the freedoms we have in the digital present. Fortunately, difficult as it is, we can still try to change some expressions of philosophy that are on the verge of becoming locked in place in the tools we use to understand one another and the world.

A Happy Surprise

The rise of the web was a rare instance when we learned new, positive information about human potential. Who would have guessed (at least at first) that millions of people would put so much effort into a project without the presence of advertising, commercial motive, threat of punishment, charismatic figures, identity politics, exploitation of the fear of death, or any of the other classic motivators of mankind. In vast numbers, people did something cooperatively, solely because it was a good idea, and it was beautiful.

Some of the more wild-eyed eccentrics in the digital world had guessed that it would happen—but even so it was a shock when it actually did come to pass. It turns out that even an optimistic, idealistic philosophy is realizable. Put a happy philosophy of life in software, and it might very well come true!

Technology Criticism Shouldn't Be Left to the Luddites

But not all surprises have been happy.

This digital revolutionary still believes in most of the lovely deep ideals that energized our work so many years ago. At the core was a sweet faith in human nature. If we empowered individuals, we believed, more good than harm would result.

The way the internet has gone sour since then is truly perverse. The central faith of the web's early design has been superseded by a different faith in the centrality of imaginary entities epitomized by the idea that the internet as a whole is coming alive and turning into a superhuman creature.

The designs guided by this new, perverse kind of faith put people back in the shadows. The fad for anonymity has undone the great opening-of-everyone's-windows of the 1990s. While that reversal has empowered sadists to a degree, the worst effect is a degradation of ordinary people.

Part of why this happened is that volunteerism proved to be an extremely powerful force in the first iteration of the web. When businesses rushed in to capitalize on what had happened, there was something of a problem, in that the content aspect of the web, the cultural side, was functioning rather well without a business plan.

Google came along with the idea of linking advertising and searching, but that business stayed out of the middle of what people actually did online. It had indirect effects, but not direct ones. The early waves of web activity were remarkably energetic and had a personal quality. People created personal "homepages," and each of them was different, and often strange. The web had flavor.

Entrepreneurs naturally sought to create products that would inspire demand (or at least hypothetical advertising opportunities that might someday compete with Google) where there was no lack to be addressed and no need to be filled, other than greed. Google had discovered a new permanently entrenched niche enabled by the nature of digital technology. It turns out that the digital system of representing people and ads so they can be matched is like MIDI. It is an example of how digital technology can cause an explosive increase in the importance of the "network effect." Every element in the system—every computer, every person, every bit—comes to depend on relentlessly detailed adherence to a common standard, a common point of exchange.

Unlike MIDI, Google's secret software standard is hidden in its computer cloud* instead of being replicated in your pocket. Anyone who wants to place ads must use it, or be out in the cold, relegated to a tiny, irrelevant subculture, just as digital musicians must use MIDI in order to work together in the digital realm. In the case of Google, the monopoly is opaque and proprietary. (Sometimes locked-in digital niches are

*"Cloud" is a term for a vast computing resource available over the Internet. You never know where the cloud resides physically. Google, Microsoft, IBM, and various government agencies are some of the proprietors of computing clouds.

proprietary, and sometimes they aren't. The dynamics are the same in either case, though the commercial implications can be vastly different.)

There can be only one player occupying Google's persistent niche, so most of the competitive schemes that came along made no money. Behemoths like Facebook have changed the culture with commercial intent, but without, as of this time of writing, commercial achievement.*

In my view, there were a large number of ways that new commercial successes might have been realized, but the faith of the nerds guided entrepreneurs on a particular path. Voluntary productivity had to be commoditized, because the type of faith I'm criticizing thrives when you can pretend that computers do everything and people do nothing.

An endless series of gambits backed by gigantic investments encouraged young people entering the online world for the first time to create standardized presences on sites like Facebook. Commercial interests promoted the widespread adoption of standardized designs like the blog, and these designs encouraged pseudonymity in at least some aspects of their designs, such as the comments, instead of the proud extroversion that characterized the first wave of web culture.

Instead of people being treated as the sources of their own creativity, commercial aggregation and abstraction sites presented anonymized fragments of creativity as products that might have fallen from the sky or been dug up from the ground, obscuring the true sources.

Tribal Accession

The way we got here is that one subculture of technologists has recently become more influential than the others. The winning subculture doesn't have a formal name, but I've sometimes called the members "cybernetic totalists" or "digital Maoists."

*Facebook does have advertising, and is surely contemplating a variety of other commercial plays, but so far has earned only a trickle of income, and no profits. The same is true for most of the other web 2.0 businesses. Because of the enhanced network effect of all things digital, it's tough for any new player to become profitable in advertising, since Google has already seized a key digital niche (its ad exchange). In the same way, it would be extraordinarily hard to start a competitor to eBay or Craigslist. Digital network architectures naturally incubate monopolies. That is precisely why the idea of the noosphere, or a collective brain formed by the sum of all the people connected on the internet, has to be resisted with more force than it is promoted.

The ascendant tribe is composed of the folks from the open culture/Creative Commons world, the Linux community, folks associated with the artificial intelligence approach to computer science, the web 2.0 people, the anticontext file sharers and remashers, and a variety of others. Their capital is Silicon Valley, but they have power bases all over the world, wherever digital culture is being created. Their favorite blogs include Boing Boing, TechCrunch, and Slashdot, and their embassy in the old country is *Wired*.

Obviously, I'm painting with a broad brush; not every member of the groups I mentioned subscribes to every belief I'm criticizing. In fact, the groupthink problem I'm worried about isn't so much in the minds of the technologists themselves, but in the minds of the users of the tools the cybernetic totalists are promoting.

The central mistake of recent digital culture is to chop up a network of individuals so finely that you end up with a mush. You then start to care about the abstraction of the network more than the real people who are networked, even though the network by itself is meaningless. Only the people were ever meaningful.

When I refer to the tribe, I am not writing about some distant "them." The members of the tribe are my lifelong friends, my mentors, my students, my colleagues, and my fellow travelers. Many of my friends disagree with me. It is to their credit that I feel free to speak my mind, knowing that I will still be welcome in our world.

On the other hand, I know there is also a distinct tradition of computer science that is humanistic. Some of the better-known figures in this tradition include the late Joseph Weizenbaum, Ted Nelson, Terry Winograd, Alan Kay, Bill Buxton, Doug Englebart, Brian Cantwell Smith, Henry Fuchs, Ken Perlin, Ben Schneiderman (who invented the idea of clicking on a link), and Andy Van Dam, who is a master teacher and has influenced generations of protégés, including Randy Pausch. Another important humanistic computing figure is David Gelernter, who conceived of a huge portion of the technical underpinnings of what has come to be called cloud computing, as well as many of the potential practical applications of clouds.

And yet, it should be pointed out that humanism in computer science doesn't seem to correlate with any particular cultural style. For instance,

Ted Nelson is a creature of the 1960s, the author of what might have been the first rock musical (*Anything @ Everything*), something of a vagabond, and a counterculture figure if ever there was one. David Gelernter, on the other hand, is a cultural and political conservative who writes for journals like *Commentary* and teaches at Yale. And yet I find inspiration in the work of them both.

Trap for a Tribe

The intentions of the cybernetic totalist tribe are good. They are simply following a path that was blazed in earlier times by well-meaning Freudians and Marxists—and I don't mean that in a pejorative way. I'm thinking of the earliest incarnations of Marxism, for instance, before Stalinism and Maoism killed millions.

Movements associated with Freud and Marx both claimed foundations in rationality and the scientific understanding of the world. Both perceived themselves to be at war with the weird, manipulative fantasies of religions. And yet both invented their own fantasies that were just as weird.

The same thing is happening again. A self-proclaimed materialist movement that attempts to base itself on science starts to look like a religion rather quickly. It soon presents its own eschatology and its own revelations about what is really going on—portentous events that no one but the initiated can appreciate. The Singularity and the noosphere, the idea that a collective consciousness emerges from all the users on the web, echo Marxist social determinism and Freud's calculus of perversions. We rush ahead of skeptical, scientific inquiry at our peril, just like the Marxists and Freudians.

Premature mystery reducers are rent by schisms, just like Marxists and Freudians always were. They find it incredible that I perceive a commonality in the membership of the tribe. To them, the systems Linux and UNIX are completely different, for instance, while to me they are coincident dots on a vast canvas of possibilities, even if much of the canvas is all but forgotten by now.

At any rate, the future of religion will be determined by the quirks of the software that gets locked in during the coming decades, just like the futures of musical notes and personhood.

Where We Are on the Journey

It's time to take stock. Something amazing happened with the introduction of the World Wide Web. A faith in human goodness was vindicated when a remarkably open and unstructured information tool was made available to large numbers of people. That openness can, at this point, be declared "locked in" to a significant degree. Hurray!

At the same time, some not-so-great ideas about life and meaning were also locked in, like MIDI's nuance-challenged conception of musical sound and UNIX's inability to cope with time as humans experience it.

These are acceptable costs, what I would call aesthetic losses. They are counterbalanced, however, by some aesthetic victories. The digital world looks better than it sounds because a community of digital activists, including folks from Xerox Parc (especially Alan Kay), Apple, Adobe, and the academic world (especially Stanford's Don Knuth) fought the good fight to save us from the rigidly ugly fonts and other visual elements we'd have been stuck with otherwise.

Then there are those recently conceived elements of the future of human experience, like the already locked-in idea of the file, that are as fundamental as the air we breathe. The file will henceforth be one of the basic underlying elements of the human story, like genes. We will never know what that means, or what alternatives might have meant.

On balance, we've done wonderfully well! But the challenge on the table now is unlike previous ones. The new designs on the verge of being locked in, the web 2.0 designs, actively demand that people define themselves downward. It's one thing to launch a limited conception of music or time into the contest for what philosophical idea will be locked in. It is another to do that with the very idea of what it is to be a person.

Why It Matters

If you feel fine using the tools you use, who am I to tell you that there is something wrong with what you are doing? But consider these points:

- ▶ Emphasizing the crowd means deemphasizing individual humans in the design of society, and when you ask people not to be people, they revert to bad moblike behaviors.

This leads not only to empowered trolls, but to a generally unfriendly and unconstructive online world.

- ▶ Finance was transformed by computing clouds. Success in finance became increasingly about manipulating the cloud at the expense of sound financial principles.
- ▶ There are proposals to transform the conduct of science along similar lines. Scientists would then understand less of what they do.
- ▶ Pop culture has entered into a nostalgic malaise. Online culture is dominated by trivial mashups of the culture that existed before the onset of mashups, and by fandom responding to the dwindling outposts of centralized mass media. It is a culture of reaction without action.
- ▶ Spirituality is committing suicide. Consciousness is attempting to will itself out of existence.

It might seem as though I'm assembling a catalog of every possible thing that could go wrong with the future of culture as changed by technology, but that is not the case. All of these examples are really just different aspects of one singular, big mistake.

The deep meaning of personhood is being reduced by illusions of bits. Since people will be inexorably connecting to one another through computers from here on out, we must find an alternative.

We have to think about the digital layers we are laying down now in order to benefit future generations. We should be optimistic that civilization will survive this challenging century, and put some effort into creating the best possible world for those who will inherit our efforts.

Next to the many problems the world faces today, debates about online culture may not seem that pressing. We need to address global warming, shift to a new energy cycle, avoid wars of mass destruction, support aging populations, figure out how to benefit from open markets without being disastrously vulnerable to their failures, and take care of other basic business. But digital culture and related topics like the future of privacy and copyrights concern the society we'll have if we can survive these challenges.

Every save-the-world cause has a list of suggestions for "what each of us can do": bike to work, recycle, and so on.

I can propose such a list related to the problems I'm talking about:

- ▶ Don't post anonymously unless you really might be in danger.
- ▶ If you put effort into Wikipedia articles, put even more effort into using your personal voice and expression outside of the wiki to help attract people who don't yet realize that they are interested in the topics you contributed to.
- ▶ Create a website that expresses something about who you are that won't fit into the template available to you on a social networking site.
- ▶ Post a video once in a while that took you one hundred times more time to create than it takes to view.
- ▶ Write a blog post that took weeks of reflection before you heard the inner voice that needed to come out.
- ▶ If you are twittering, innovate in order to find a way to describe your internal state instead of trivial external events, to avoid the creeping danger of believing that objectively described events define you, as they would define a machine.

These are some of the things you can do to be a person instead of a source of fragments to be exploited by others.

There are aspects to all these software designs that could be retained more humanistically. A design that shares Twitter's feature of providing ambient continuous contact between people could perhaps drop Twitter's adoration of fragments. We don't really know, because it is an unexplored design space.

As long as you are not defined by software, you are helping to broaden the identity of the ideas that will get locked in for future generations. In most arenas of human expression, it's fine for a person to love the medium they are given to work in. Love paint if you are a painter; love a

clarinet if you are a musician. Love the English language (or hate it). Love of these things is a love of mystery.

But in the case of digital creative materials, like MIDI, UNIX, or even the World Wide Web, it's a good idea to be skeptical. These designs came together very recently, and there's a haphazard, accidental quality to them. Resist the easy grooves they guide you into. If you love a medium made of software, there's a danger that you will become entrapped in someone else's recent careless thoughts. Struggle against that!

The Importance of Digital Politics

There was an active campaign in the 1980s and 1990s to promote visual elegance in software. That political movement bore fruit when it influenced engineers at companies like Apple and Microsoft who happened to have a chance to steer the directions software was taking before lock-in made their efforts moot.

That's why we have nice fonts and flexible design options on our screens. It wouldn't have happened otherwise. The seemingly unstoppable mainstream momentum in the world of software engineers was pulling computing in the direction of ugly screens, but that fate was avoided before it was too late.

A similar campaign should be taking place now, influencing engineers, designers, businesspeople, and everyone else to support humanistic alternatives whenever possible. Unfortunately, however, the opposite seems to be happening.

Online culture is filled to the brim with rhetoric about what the true path to a better world ought to be, and these days it's strongly biased toward an antihuman way of thinking.

The Future

The true nature of the internet is one of the most common topics of online discourse. It is remarkable that the internet has grown enough to contain the massive amount of commentary about its own nature.

The promotion of the latest techno-political-cultural orthodoxy, which I am criticizing, has become unceasing and pervasive. The *New York*

Times, for instance, promotes so-called open digital politics on a daily basis even though that ideal and the movement behind it are destroying the newspaper, and all other newspapers.* It seems to be a case of journalistic Stockholm syndrome.

There hasn't yet been an adequate public rendering of an alternative worldview that opposes the new orthodoxy. In order to oppose orthodoxy, I have to provide more than a few jabs. I also have to realize an alternative intellectual environment that is large enough to roam in. Someone who has been immersed in orthodoxy needs to experience a figure-ground reversal in order to gain perspective. This can't come from encountering just a few heterodox thoughts, but only from a new encompassing architecture of interconnected thoughts that can engulf a person with a different worldview.

So, in this book, I have spun a long tale of belief in the opposites of computationalism, the noosphere, the Singularity, web 2.0, the long tail, and all the rest. I hope the volume of my contrarianism will foster an alternative mental environment, where the exciting opportunity to start creating a new digital humanism can begin.

An inevitable side effect of this project of deprogramming through immersion is that I will direct a sustained stream of negativity onto the ideas I am criticizing. Readers, be assured that the negativity eventually tapers off, and that the last few chapters are optimistic in tone.

*Today, for instance, as I write these words, there was a headline about R, a piece of geeky statistical software that would never have received notice in the *Times* if it had not been "free." R's nonfree competitor Stata was not even mentioned. (Ashlee Vance, "Data Analysts Captivated by R's Power," *New York Times*, January 6, 2009.)