"Gershenfeld's account of the technology's evolution is delicious. A star physicist at MIT with a knack for technical explanation, he has written an accessible book that even non-technophiles will love." —*BUSINESSWEEK* (Best Book of the Year selection)

FAB

THE COMING REVOLUTION ON YOUR DESKTOP—FROM PERSONAL COMPUTERS TO PERSONAL FABRICATION

NEIL GERSHENFELD
AUTHOR OF
WHEN THINGS START TO THINK

"Gershenfeld's account of the technology's evolution is delicious. A star physicist at MIT with a knack for technical explanation, he has written an accessible book that even non-technophiles will love." —*BUSINESSWEEK* (Best Book of the Year selection)

FAB

THE COMING REVOLUTION ON YOUR DESKTOP—FROM PERSONAL COMPUTERS TO PERSONAL FABRICATION

NEIL GERSHENFELD
AUTHOR OF
WHEN THINGS START TO THINK

Table of Contents

The Future

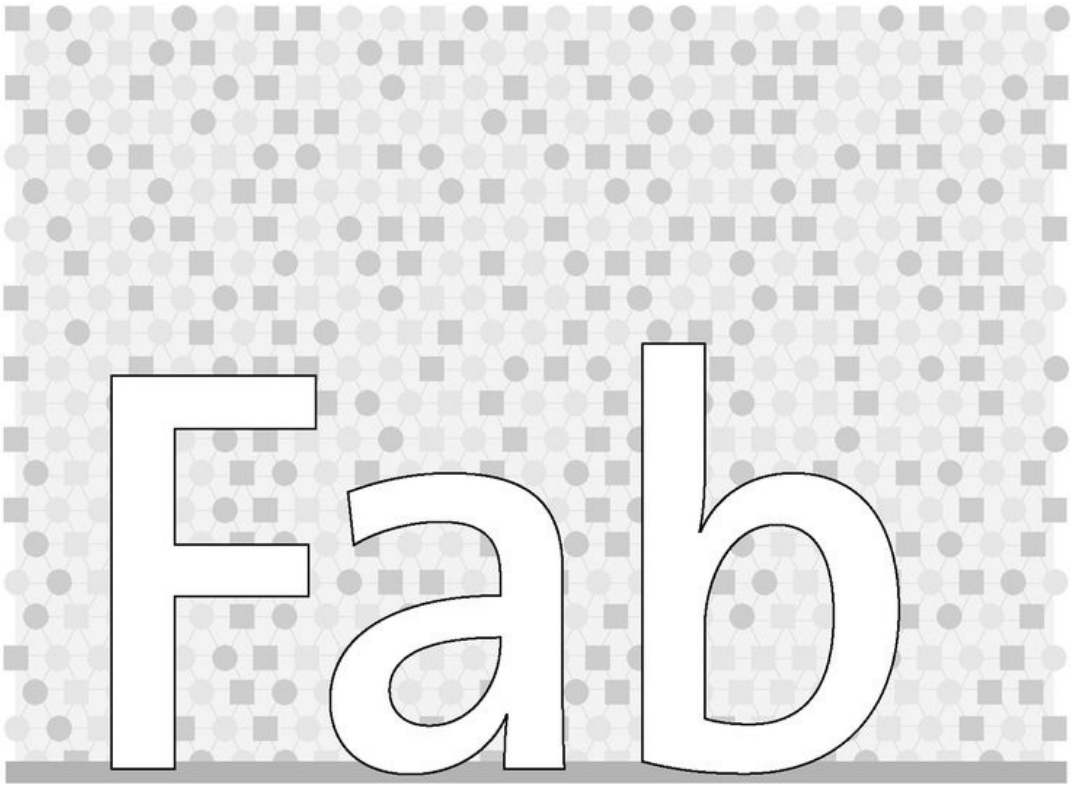
The Details

Also by Neil Gershenfeld

When Things Start to Think

The Nature of Mathematical Modeling

The Physics of Information Technology



The Coming Revolution on Your Desktop—from Personal Computers to Personal Fabrication

Neil Gershenfeld

BASIC BOOKS
A MEMBER OF THE PERSEUS BOOKS GROUP
NEW YORK

For Kalbag

and

Kelly, Meejin, Dalia, Irene, Saul, Alan, Grace,
Eli, Mel, Kyei, Anil, Frank, Larry, Etienne, Seymour,
David, Kenny, Amy, Haakon, Vicente, Terry,
Sugata, Arjun

and

Lass, Susan, John, Bakhtiar, Amon, Joe, Aisha,
Chris, Caroline, Manu, Sanjay, Debu,
Jørgen, Benn, Milton, Joe, Ike, Scott, Seth,
Alex, Mitch, Marvin

How to Make . . .

Mainframe computers were expensive machines with limited markets, used by skilled operato working in specialized rooms to perform repetitive industrial operations. We can laugh in retrospect the small size of the early sales forecasts for computers; when the packaging of computation made accessible to ordinary people in the form of personal computers, the result was a unprecedente outpouring of new ways to work and play.

However, the machines that make computers (and most everything else) remain expensive too with limited markets, used by skilled operators working in specialized rooms to perform repetitiv industrial operations. Like the earlier transition from mainframes to PCs, the capabilities of machi tools will become accessible to ordinary people in the form of personal fabricators (PFs). This tim around, though, the implications are likely to be even greater because what's being personalized is o physical world of atoms rather than the computer's digital world of bits.

A PF is a machine that makes machines; it's like a printer that can print *things* rather than image By personal fabrication, I mean not only the creation of three-dimensional structures but also th integration of logic, sensing, actuation, and display—everything that's needed to make a comple functioning system. With a PF, instead of shopping for and ordering a product, you could download d develop its description, supplying the fabricator with designs and raw materials.

Programmable personal fabricators are not just a prediction, they're a reality. The world o tomorrow can be glimpsed in tools available today. *Fab* tells the stories of these remarkable tools an their equally remarkable users around the world. It explains what can be made, and why, and how.

I first encountered the possibility of personal fabrication through the unexpectedly enthusiast student response to a class that I teach at MIT, modestly titled "How To Make (almost) Anything." A MIT I direct the Center for Bits and Atoms. CBA comprises fifteen or so faculty from across campu physicists, chemists, biologists, mathematicians, and mechanical and electrical engineers. They al like me, never fit into the artificial separation of computer science from physical science.

The universe is literally as well as metaphorically a computer. Atoms, molecules, bacteria, an billiard balls can all store and transform information. Using the discrete language of computatio rather than the continuous equations of calculus to describe the behavior of physical systems is n only leading to the practical development of new and more powerful kinds of informatio technologies, such as quantum computers, it's also leading to new kinds of insights into the nature o the universe itself, such as the long-term behavior of black holes. If the world is a computer, then th science of computing is really the science of science.

At the intersection of physical science and computer science, programs can process atoms as well bits, digitizing fabrication in the same way that communications and computation were earli digitized. Ultimately, this means that a programmable personal fabricator will be able to mak anything, including itself, by assembling atoms. It will be a self-reproducing machine. That idea h been a long-standing science fiction staple for better or, sometimes, much worse.

In *Star Trek: The Next Generation*, the replicator is an essential plot element that is capable making whatever is needed for each episode. It looks like an overgrown drinks dispenser, but it has th useful feature of being able to dispense anything. In theory, it does this by following store instructions to put together subatomic particles to make atoms, atoms to make molecules, an

molecules to make whatever you want. For Captain Picard, that was frequently a steaming mug of h
preferred tea, obtained from the replicator with the command "Tea, Earl Grey, hot."

The less fortunate Arthur Ford in the *Hitchhiker's Guide to the Galaxy* had to contend with th
infamous Nutri-Matic machine to obtain his cup of tea. Rather than storing the molecul
specification in advance, the Nutri-Matic attempted to personalize Arthur's beverage by performing
spectroscopic analysis of his metabolism, and then probing the taste centers in his brain. As wi
Captain Picard's tea, Arthur's drink is synthesized by assembling its molecular constituents. Howeve
in the Nutri-Matic's case the inevitable result was a plastic cup filled with a liquid that was almos
but not quite, entirely unlike tea.

None of this violates any physical laws, and in fact such atomic-scale programmable assembly
already possible in the lab today (as long as your tastes don't run to anything much larger than a fe
atoms).

To develop real working personal fabricators that can operate on a larger scale, my colleagues
MIT and I assembled an array of machines to make the machines that make machines. These too
used supersonic jets of water, or powerful lasers, or microscopic beams of atoms to make—wel
almost anything. The problem we quickly ran into was that it would take a lifetime of classes f
students to master all of the tools, and even then the students would get little practical experience
combining these tools to create complete working systems. So, we thought, why not offer a singl
semester course that would provide a hands-on introduction to all the machines?

In 1998 we tried teaching "How To Make (almost) Anything" for the first time. The course wa
aimed at the small group of advanced students who would be using these tools in their researc
Imagine our surprise, then, when a hundred or so students showed up for a class that could hold on
ten. They weren't the ones we expected, either; there were as many artists and architects as engineer
And student after student said something along the lines of "All my life I've been waiting to take
class like this," or "I'll do anything to get into this class." Then they'd quietly ask, "This seems to b
too useful for a place like MIT—are you really allowed to teach it here?"

Students don't usually behave that way. Something had to be wrong with this class, or with all th
other classes I taught. I began to suspect the latter.

The overwhelming interest from students with relatively little technical experience (for MIT) wa
only the first surprise. The next was the reason why they wanted to take the class. Virtually no one wa
doing this for research. Instead, they were motivated by the desire to make things they'd alway
wanted, but that didn't exist. These ranged from practical (an alarm clock that needs to be wrestle
into turning off), to fanciful (a Web browser for parrots), to profoundly quirky (a portable person
space for screaming). Their inspiration wasn't professional; it was personal. The goal was not
publish a paper, or file a patent, or market a product. Rather, their motivation was their own pleasu
in making and using their inventions.

The third surprise was what these students managed to accomplish. Starting out with skills mo
suited to arts and crafts than advanced engineering, they routinely and single-handedly managed
design and build complete functioning systems. Doing this entailed creating both the physical form-
mastering the use of computer-controlled tools that produce three-dimensional shapes by adding
removing material—and the logical function—designing and building circuits containing embedde
computer chips interfaced with input and output devices. In an industrial setting these tasks a
distributed over whole teams of people who conceive, design, and produce a product. No one memb
of such a team could do all of this, and even if they could, they wouldn't: personal screamin
technology is unlikely to emerge as a product plan from a marketing meeting (even if the participan

might secretly long for it).

The final surprise was how these students learned to do what they did: the class turned out to l something of an intellectual pyramid scheme. Just as a typical working engineer would not have th design and manufacturing skills to personally produce one of these projects, no single curriculum teacher could cover the needs of such a heterogeneous group of people and machines. Instead, th learning process was driven by the demand for, rather than supply of, knowledge. Once studen mastered a new capability, such as waterjet cutting or microcontroller programming, they had a nea evangelical interest in showing others how to use it. As students needed new skills for their projec they would learn them from their peers and then in turn pass them on. Along the way, they woul leave behind extensive tutorial material that they assembled as they worked. This phase might last month or so, after which they were so busy using the tools that they couldn't be bothered to docume anything, but by then others had taken their place. This process can be thought of as a "just-in-time educational model, teaching on demand, rather than the more traditional "just-incase" model th covers a curriculum fixed in advance in the hopes that it will include something that will later l useful.

These surprises have recurred with such certainty year after year that I began to realize that the students were doing much more than taking a class; they were inventing a new physical notion literacy. The common understanding of "literacy" has narrowed down to reading and writing, b when the term emerged in the Renaissance it had a much broader meaning as a mastery of th available means of expression. However, physical fabrication was thrown out as an "illiberal art pursued for mere commercial gain. These students were correcting a historical error, using millions dollars' worth of machinery for technological expression every bit as eloquent as a sonnet or painting.

Today there aren't many places where these kinds of tools are available for play rather than wor but their capabilities will be integrated into accessible and affordable consumer versions. Such future really represents a return to our industrial roots, before art was separated from artisans, wh production was done for individuals rather than masses. Life without the infrastructure we take f granted today required invention as a matter of survival rather than as a specialized profession. Th design, production, and use of engineered artifacts—agricultural implements, housewares, weapon and armor—all took place locally. The purpose of bringing tool-making back into the home is not recreate the hardships of frontier living, just as it's not to run personalscream-container productio lines out of the family room. Rather, it's to put control of the creation of technology back in the hand of its users.

The analogy between the personalization of fabrication and computation is close, and instructiv Remember that mainframes were behemoths; in 1949 *Popular Mechanics* famously forecast th "Computers in the future may weigh no more than 1.5 tons." The Digital Equipment Corporatio (DEC) pioneered computers that were the size of desks rather than rooms. They called the "Programmed Data Processors" (PDPs) rather than computers because the market for computers w seen as being too small to be viable. But over time this class of devices came to be calle *minicomputers*. A 1964 DEC ad proudly proclaimed, "Now you can own the PDP-5 computer for wh a core memory alone used to cost: $27,000." That's a lot more than a PC costs today, but a lot les than the price of a mainframe. Minicomputers were thus accessible to small groups of users rath than just large corporations, and consequently their uses migrated from meeting the needs corporations to satisfying individuals. PDPs did eventually shrink down so that they could fit on desk, but they didn't end up there, because the engineers developing them didn't see wh

nonengineers would want them. DEC's president, Ken Olsen, said in 1977 that "there is no reason f any individual to have a computer in their home." PCs are now in the home; DEC is now defunct.

The adoption of PCs was driven by "killer apps," applications that were so compelling the motivated people to buy the systems to run them. The classic killer app was the original spreadshee VisiCalc, which in 1979 turned the Apple II from a hobbyist's toy to a serious business tool, an helped propel IBM into the PC business. VisiCalc's successor, Lotus 1-2-3, did the same in 1983 fo IBM's PCs. The machine tools that the students taking "How To Make (almost) Anything" use toda are much like mainframes, filling rooms and costing hundreds of thousands of dollars. But despi their size and expense, they've been adequate to show, year in and year out, that the killer app fo personal fabrication is fulfilling individual desires rather than merely meeting mass-market needs. F one student this meant putting a parrot online; for another an alarm clock that got her up in th morning. None of the students needed to convince anyone else of the value of their ideas; they ju created them themselves.

The invention that made PCs possible was integrated circuits, leading up to the development microprocessors that put the heart of a computer onto a single silicon chip. The invention th promises to put the capabilities of a roomful of machine tools onto a desktop is the printing functional materials. A printer's ink-jet cartridge contains reservoirs of cyan, magenta, yellow, an black inks; by precisely placing tiny drops it's possible to mix these to produce what appear to b perfect reproductions of any image. In the research lab today, there are similar inks that can be used print insulators, conductors, and semiconductors to make circuits, as well as structural materials th can be deposited to make three-dimensional shapes. The integration of these functional materials in a printing cartridge will make possible faithful reproductions of arbitrary objects as well as images. A MIT we have a joke that we now take seriously: a student working on this project can graduate whe their thesis can walk out of the printer. In other words, along with printing the text of the documen the printer must produce the means for the thesis to move.

Ultimately, rather than relying on a printer to place droplets of material, the logic for assembling a object will be built into the materials themselves. This is exactly how our bodies are made; molecular machine called the ribosome translates instructions from genes into the series of step required to assemble all of the proteins in our bodies out of the twenty amino acids. The discovery building with logic is actually a few billion years old; it's fundamental to the emergence of lif Current research is now seeking to do the same with functional materials, creating a fundamental digital fabrication process based on programming the assembly of microscopic building blocks. Th mechanism will be embodied in personal fabricators fed by such structured materials. Much as machine today might need supplies of air, water, and electricity, a digital personal fabricator will us as raw feedstocks streams of conductors, semiconductors, and insulators.

Unlike machines of today, though, but just like a child's building blocks, personal fabricators wi also be able to disassemble something and sort its constituents, because the assembled objects a constructed from a fixed set of parts. The inverse of digital fabrication is digital recycling. An obje built with digital materials can contain enough information to describe its construction, and hence i deconstruction, so that an assembler can run in reverse to take it apart and reuse its raw materials.

We're now on the threshold of a digital revolution in fabrication. The earlier revolutions i digitizing communications and computation allowed equipment made from unreliable components reliably send messages and perform computations; the digitization of fabrication will allow perfe macroscopic objects to be made out of imperfect microscopic components, by correcting errors in th assembly of their constituents.

Return now to the mainframe analogy. The essential step between mainframes and PCs wa minicomputers, and a similar sequence is happening along the way to personal fabrication. It possible to approximate the end point of that evolution today with a few thousand dollars equipment on a desktop, because engineering in space and time has become cheap.

First the space part. An inexpensive CD player places its read head with a resolution of a millionth of a meter, a micron. Combining this metrology with a desktop computer-controlled milling machin enables it to move a cutting tool in three dimensions with that same resolution, patterning shapes b removing material with tolerances approaching the limits of human perception. For example, it ca cut out circuit boards with features as fine as the tiniest components.

Then the time part. A one-dollar embedded computer chip can operate faster than a millionth of second, a microsecond. This is fast enough to use software to perform functions that traditionall required custom hardware, such as generating communications signals and controlling displays. It possible to program one such chip to take on the functions of many different kinds of circuits.

The increasing accessibility of space and time means that a relatively modest facility (on the sca of the MIT class) can be used to create physical forms as fine as microns and program logic functions as fast as microseconds. Such a lab needs more complex consumables than the ink requir by a printer, including copper-clad boards to make circuits and computer chips to embed into project But, as the students found at MIT, these capabilities can be combined to create complete functionin systems. The end result is much the same as what an integrated PF will be able to do, allowin technological invention by end users.

Minicomputers showed how PCs would ultimately be used, from word processing to e-mail to th Internet, long before technological development caught up to make them cheap enough and simp enough for wider adoption. Struck by this analogy, I wondered if it would be possible to deploy prot personal fabricators in order to learn now about how they'll be used instead of waiting for all of th research to be completed.

This thought led to the launch of a project to create field "fab labs" for exploring the implicatio and applications of personal fabrication in those parts of the planet that don't get to go to MIT. As yo wish, "fab lab" can mean a lab for fabrication, or simply a fabulous laboratory. Just as a minicomput combined components—the processor, the tape drive, the keypunch, and so forth—that we originally housed in separate cabinets, a fab lab is a collection of commercially available machin and parts linked by software and processes we developed for making things. The first fab labs have laser cutter to cut out two-dimensional shapes that can be assembled into three-dimension structures, a sign cutter that uses a computer-controlled knife to plot flexible electrical connectio and antennas, a milling machine that moves a rotating cutting tool in three dimensions to make circu boards and precision parts, and the tools for programming tiny high-speed microcontrollers to embe logic. A bit like the original PDP minicomputers, all of this could be called a Programmed Materia Processor. This is not a static configuration; the intention over time is to replace parts of the fab la with parts made in the fab lab, until eventually the labs themselves are self-reproducing.

The National Science Foundation (NSF) provided the seed funding for fab labs through its suppo of the Center for Bits and Atoms (CBA). NSF expects research activities that it funds on the scale o CBA to have an educational outreach component, which all too often is limited to teaching som classes at a local school, or creating a Web site describing the research. Instead, my CBA colleague and our NSF counterparts agreed to try equipping ordinary people to actually *do* what we're studyin at MIT instead of just talking about it. It's possible in the fab labs to do work that not too long ag required the resources of a place like MIT (an observation that's not lost on me on days when th

administrative overhead of working at a place like MIT is particularly onerous).

Starting in 2002, the first fab labs went to rural India, Costa Rica, northern Norway, inner-ci Boston, and Ghana. The equipment and supplies for each site initially cost about twenty thousar dollars. Knowing that that cost will come down as the technology progresses, the first fab labs were meant to be economically self-sustaining. One of the first surprises from the field was the demand f duplicating the labs even at that cost.

In keeping with the fab lab project's goal of discovering which tools and processes would be mo useful in the field, we started setting up these labs long before we knew how best to do it. Th response in the field was as immediate as it had been at MIT. We ended up working in so many fa flung locations because we found a demand for these capabilities around the world that was every b as strong as that around campus. In the village of Pabal in western India, there was interest in usi the lab to develop measurement devices for applications ranging from milk safety to agricultur engine efficiency. In Bithoor, on the bank of the Ganges, local women wanted to do three-dimension scanning and printing of the carved wooden blocks used for chikan, a local kind of embroidery. Sam herders in the Lyngen Alps of northern Norway wanted wireless networks and animal tags so that the data could be as nomadic as their animals. People in Ghana wanted to create machines direct powered from their abundant sunlight instead of scarce electricity. Children in inner-city Boston use their fab lab to turn scrap material into sellable jewelry.

For all the attention to the "digital divide" in access to computers between developed an developing countries, these recurring examples suggest that there is an even more significant divide access to tools for fabrication and instrumentation. Desktop computers are of little use in places th don't have desks; all too often they sit idle in isolated rooms built by aid agencies to house them Appropriate computing requires the means to make, measure, and modify the physical world of aton as well as the virtual world of bits. And instead of bringing information technology (IT) to the masse fab labs show that it's possible to bring the tools for IT development, in order to develop and produc local technological solutions to local problems.

There's been a long-standing bias that technology's role in global development mirrors its ow history, progressing from low- to high-tech. The fab lab experience suggests instead that some of th least developed parts of the world need some of the most advanced technologies. That observation h led me to spend some head-spinning days in Washington, DC, going from the World Bank to th National Academy of Sciences to Capitol Hill to the Pentagon, having essentially the same meeting each place. Fab labs challenge assumptions that are fundamental to each of these institutions. Instea of spending vast sums to send computers around the world, it's possible to send the means to mak them. Instead of trying to interest kids in science as received knowledge, it's possible to equip them *do* science, giving them both the knowledge and the tools to discover it. Instead of building bett bombs, emerging technology can help build better communities.

The problem I found on these trips was that none of these institutions knew how to pay for this kin of work; there isn't a Pentagon Office of Advanced Technologies for Avoiding Wars. Financin personal fabrication in underserved communities is too directed a goal for traditional basic researc funding, and too speculative for conventional aid organizations or donors. The closest precedent microcredit lending, which provides small loans to help support financial cooperatives, typically ru by women, in developing countries. The loans are used to acquire an asset such as a cell phone th can then be used to generate income. But that model doesn't help when the financing is for a invention. What's needed are the skills of a good venture capitalist rather than a banker. That's not a oxymoron; the best venture capitalists add value to their investments by helping shepherd, prune, an

protect ideas, build operational teams, and develop business models.

The historical parallel between personal computation and personal fabrication provides a guide what those business models might look like. Commercial software was first written by and for b companies, because only they could afford the mainframe computers needed to run it. When PCs can along anyone could become a software developer, but a big company was still required to develop an distribute big programs, notably the operating systems used to run other programs. Finally, th technical engineering of computer networks combined with the social engineering of human networ allowed distributed teams of individual developers to collaborate on the creation of the most comple software.

Programmers write source code that people can understand, which gets turned into executable cod that computers can understand. Commercial companies have protected the former and distributed th latter to their customers. But individuals who share the source code that they write can collaborate ad hoc groups, which might never meet physically, on the creation of programs that are larger tha any one of the members could write alone. The Linux operating system is built out of such "ope source" software. Much like the way science progresses by researchers building on one another publications, a programmer can make available a piece of code that might then get taken up an improved by someone on the opposite end of the earth.

In a world of open-source software, ownership of neither computers nor code alone provides th basis for a proprietary business model; what's left is the value added to them by creating content an delivering services. Profitable old and new computer companies are making money from free available software in just this way: by charging for their role in solving problems.

Similarly, possession of the means for industrial production has long been the dividing line betwee workers and owners. But if those means are easily acquired, and designs freely shared, then hardwa is likely to follow the evolution of software. Like its software counterpart, open-source hardware starting with simple fabrication functions, while nipping at the heels of complacent companies th don't believe that personal fabrication "toys" can do the work of their "real" machines. That bounda will recede until today's marketplace evolves into a continuum from creators to consumers, servicin markets ranging from one to one billion.

That transition has already happened in the case of two-dimensional printers, the immedia predecessor to personal fabricators. High-quality printing, once solely a commercial service, can into the home via laser printers. The most important attribute of an industrial printing press is i throughput, which is the number of pages it can produce per minute. Laser printers started going dow this technological scaling curve with the development of ever-faster printers needed for serious o demand commercial printing. Within Hewlett-Packard, a competing group of engineers had the ide that squirting individual drops of ink could make beautiful images more cheaply than transferrin toner onto sheets. Ink-jet printing would be slower than laser printing, but they reasoned that for printer in the home, quality mattered much more than speed. This was such a heretical idea that th group decamped from HP's headquarters in Palo Alto and set up shop out of sight in Corvalli Oregon. The rest is business history; the relative cost of producing and selling ink-jet cartridges h been about the closest a company has come to legally printing money. Ink-jet printing hasn't replace commercial printing; it's created an entirely new—and enormous—market driven by quality an access rather than speed.

Similarly, the emerging personal fabrication tools I've been describing are intended for person rather than mass production. Their development was originally driven in industry by the need quickly create prototypes of products to catch errors before they became much more expensive

correct in production. Machine-tool shows relegate such rapid-prototyping machines to a sleep corner away from the giant cutting, stamping, and molding tools that are at the top of the machin food chain. But if the market is just one person, then the prototype *is* the product. The big machin will continue to mass-produce things used in large quantities; nuts and bolts are valuable becau they're identical rather than unique. But little machines will custom-make the products that depend o differences, the kinds of things being made in the fab class and fab labs.

The biggest impediment to personal fabrication is not technical; it's already possible to effectivel do it. And it's not training; the just-in-time peer-to-peer project-based model works as well in th field as at MIT. Rather, the biggest limitation is simply the lack of knowledge that this is eve possible. Hence this book.

Fab tells the stories of pioneering personal fabrication users, and the tools they're using. Becau both are so striking, I've interwoven their stories in pairs of chapters that explore emergi applications and the processes that make them possible. The not-so-hidden agenda is to describe n only who is doing what but also how, providing introductions to the tools that are similar to th orientations we give at MIT and in the field. These stop just short of hands-on training; a final sectio gives enough detail on the products, programs, and processes used to duplicate what's shown in th book.

Throughout *Fab* I use what are known as "hello world" examples. In 1978, the instruction manu for the then new C programming language written at Bell Labs used as an example a simple progra that printed out the words "hello world." This is more exciting than it sounds, because it requires a understanding of how to write a rudimentary program, compile it into computer code, and cause th program to print the text. "Hello world" programs have since become staples for introducing ne computer languages; the Association for Computing Machinery currently lists 204 examples fo languages from A+ to zsh.

The difference between those "hello world" programs and the examples I give in this book is th mine arrange atoms as well as bits, moving material as well as instructions. But the principle is th same: show the minimum specification needed to get each of the tools to demonstrate its successf operation. Taken together, these examples provide a fairly complete picture of the means for almo anyone to make almost anything.

My hope is that *Fab* will inspire more people to start creating their own technological future We've had a digital revolution, but we don't need to keep having it. Personal fabrication will bring th programmability of the digital worlds we've invented to the physical world we inhabit. While armi of entrepreneurs, engineers, and pundits search for the next killer computer application, the bigge thing of all coming in computing lies quite literally out of the box, in making the box.

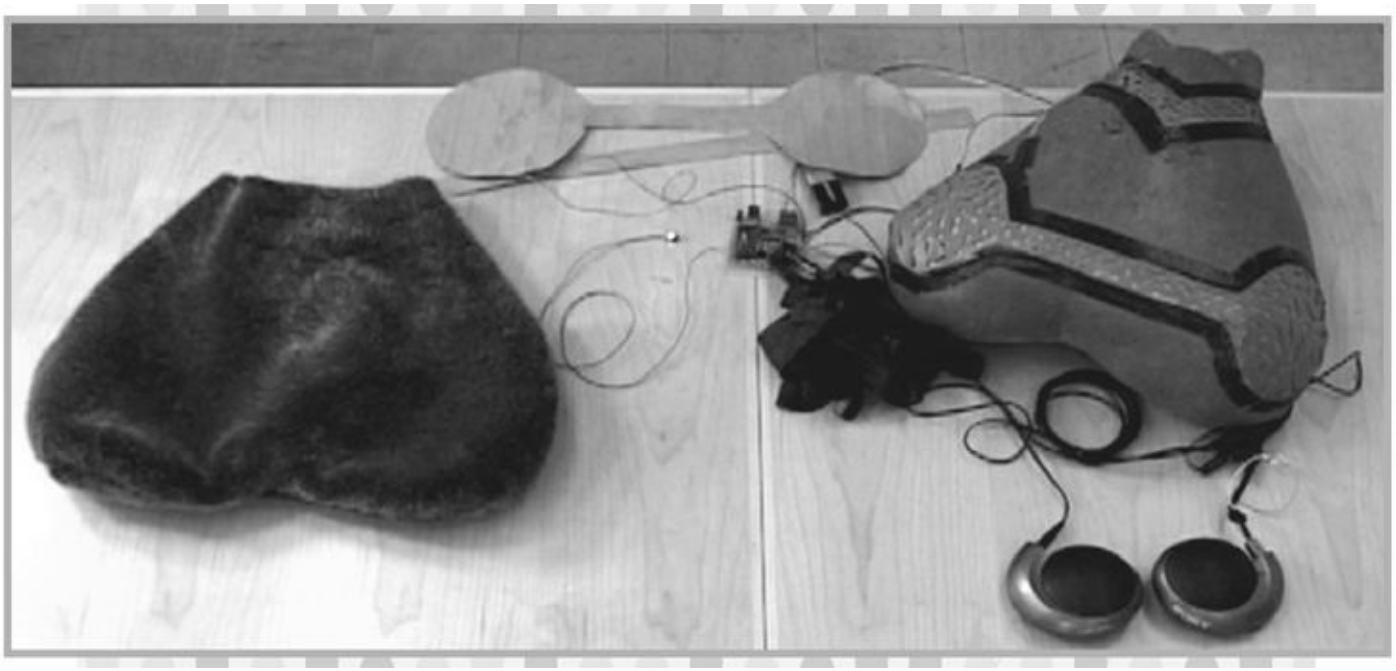
Kelly

The first year that we taught “How To Make (almost) Anything” at MIT, Kelly Dobson was the sta student. She was an artist, with lots of ideas for projects but little electronics background. Like th other students in the class, she applied the skills she had learned to her semester project. Unlike th others, though, she chose to focus on the problem of needing to scream at an inconvenient time. Th result was Kelly’s ScreamBody, which looks like a fuzzy backpack worn in front rather than in back.

As Kelly describes it during a demonstration: “Have you ever really wanted to scream? But yo can’t, because you’re at work, or in any number of situations where it’s just not appropriate? Wel ScreamBody is a portable personal space for screaming. As a person screams into ScreamBody the scream is silenced, but it is also recorded for later release where, when, and how they choose.”

There’s a familiar sequence of reactions when people hear this. First, shock: Is she really making machine to save screams? Then amusement, when they see that it actually works. And then more ofte than not they express a desire to get one of their own. Finally comes the appreciation of what Kel has in fact accomplished in making ScreamBody work.

ScreamBody



Inside ScreamBody

The first thing she did was design a circuit to save screams, then she built that into a circuit boar This in turn required programming an embedded computer to control the sound recording ar playback. Next she developed sensors for the screamer to interact with the ScreamBody, and finall assembled the apparatus into a comfortable, wearable package that could silence and later emit th scream.

In a conventional corporation, a strategy group would come up with the ScreamBody specifications, electrical engineers would design the circuits, computer scientists would program th chips, mechanical engineers would make the structures, industrial designers would assemb everything, industrial engineers would plan the production, and then marketers would try to fir someone to buy it. The only thing that Kelly finds remarkable about doing all of this herself is th anyone finds it remarkable. For Kelly, literacy means knowing how to communicate by using all th representations that information can take. She sees the design of circuits and packaging as aspects personal expression, not product development. She didn't design ScreamBody to fill a market nee she did it because she wanted one.

Unlike personal computers, or personal digital assistants, which are designed by teams of enginee to meet the common needs of the largest possible market, Kelly's personal screaming assistant targeted at a market base of one user: Kelly. What makes her exceptional is that, like most everyon else, she's not average. For her, appropriate technology helps her scream; for others it might hel them to grow food or play games or customize jewelry. Individual needs are unlikely to be met b products aimed at mass markets. A truly personal computing device is by definition not mas produced, or even mass-customized; it is personally designed.

Meejin

In the years since that first class I've been delighted to continue to find that students—ranging fro

undergrads to faculty members—have wildly different and equally quirky visions of personal appropriate technologies. Kelly was followed by Meejin Yoon, a new professor in the architectu department who like Kelly had no technical fabrication experience. Meejin took the class because sh wanted to be able to make her designs herself.

Meejin was struck by the ways that technology is encroaching on our personal space, and wonder if instead it could help to define and protect it. She expressed that thought in the Defensible Dres which was inspired by the approach of the porcupine and blowfish to protecting their personal spac Her device looks like a fringed flapper dress, but the fringes are stiff piano wire connected to the dre by motors controlled by proximity sensors. When someone comes within a distance determined by th wearer, the wires spring out to announce and defend the space around it. Countless commerci wearable computers are aimed at maximizing communication; this is one that aims to minimize it.

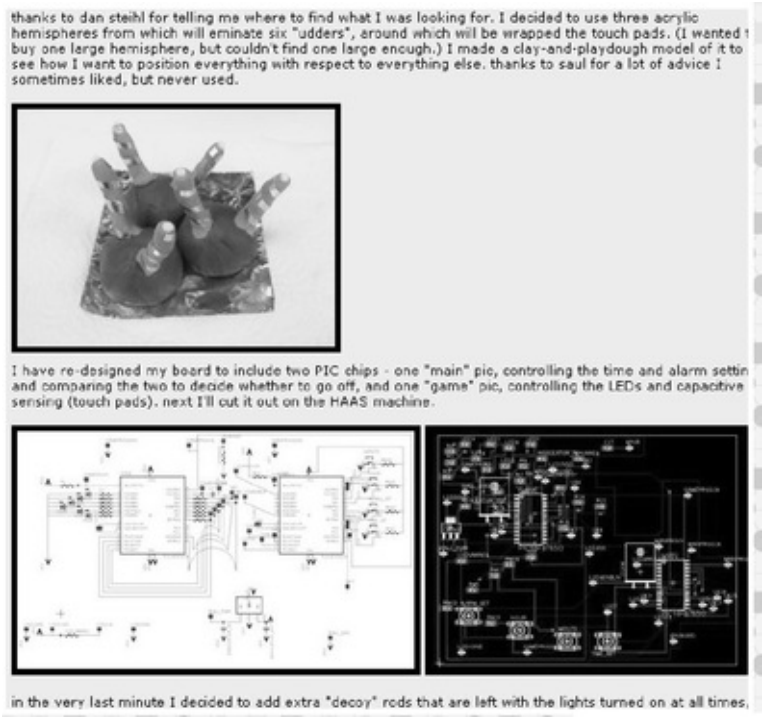
Defensive dressing

Shelly

After Meejin came Shelly Levy-Tzedek, a budding biologist who was unfamiliar with how to bui with things bigger than molecules. She chose to focus on one of the most pressing practical concer for an MIT student: waking up. Alarm clocks now on the market are too easily defeated by slee deprived scientists.

Shelly decided to make the match more even by raising the IQ of the alarm clock, so that it cou present a challenge to the would-be sleeper. She came up with an alarming-looking clock; instead simply turning off a button or activating a snooze mode, Shelly's clock demands that the user gra glowing protuberances in the order in which they randomly flash. That's a challenging enough gam when you're awake, but it becomes fiendishly difficult when you're fumbling about half-awake in th dark. When Shelly displayed the clock at a reception for fab class projects, the universal reaction w "Where can I get one?"

An alarming clock



Fab log