COMPUTERS AND PEOPLE:

PERSONAL COMPUTATION

Sherry Turkle

November 10, 1980

This report was prepared with the support of
the National Science Foundation Grant No. MCS77-27860

Key Words: Personal Computing
Computer Culture
Societal Implications of Computers

This blank page was inserted to preserve pagination.

Computers & People:

Personal Computation

by Sherry Turkle
Associate Professor of Sociology
Massachusetts Institute of Technology

In the January 1975 issue of Popular Electronics, MITS, short for Micro Instrumentation and Telemetry Systems, a small computer company in Albequerque, New Mexico, announced the Altair, a computer small enough to sit on a desktop, powerful enough to support high level language programming, and that you could build for only $420. The Altair, aimed at a "hobby" market, was followed by other small systems, marketed to less specialized audiences of small businessmen and curious householders. By 1977, Radio Shack had announced its TRS-80, a full size "turnkey" computer system that you could buy a your local store for $100 down and that would arrive fully assembled for $599. It was a year in which it was estimated that 30 million American families had stereo systems that cost that much. Computers were now no longer the exclusive property of big business, big banks, research laboratories, and electronic buffs who knew where to scrounge for used parts with the help of inside industry contacts.

Five years have passed since the announcement of the Altair and there has been a lot of computer buying in America. In 1979 the Wall Street Journal carried the estimate that there were approximately 600,000 personal computer systems in American homes. Most estimates are lower, but the number certainly has six figures. With the presence of that much hardware has come a lot of rhetoric about a personal computer revolution. Most of the talk, both from the companies that are marketing the machines and from those who claim to be the most visionary spokesmen for the people who are buying them, is about all of the things that the computer can do for you, like teach you French verbs or elementary algebra, help you with income tax and household management.

Given that industry, advertising, and computer utopians have introduced people to personal computers the way one might have introduced Aladdin to his lamp, it is not surprising that evaluations of the importance of personal computers for society and for the individual, take performance as their starting point. For example, on May 14, 1979, the Wall Street Journal reported on its own little "evaluation experiment." The paper had drafted one of its staff reporters, Mitchell Lynch, into the ranks of the home computer revolution. It presented him with a TRS-80 and told him to take it home. His assignment was to report back six months later on what it had been like. Lynch's story ran under the header: "Computer Error: Trying to Use One in Your Own Home. . . Our Man Finds that he Can't Get It to Do Tax, Other Jobs..."

The article itself is a Pilgrim's Progress of little progress. This Everyman had a lot of trouble.

> I got a spiffy $599 home computer for Christmas. And it stopped playing blackjack with me by New Year's Day, asked me "WHAT?" about 1,430 times by George Washington's birthday, and mysteriously broke down and resurrected itself by Easter.
>
> I thought it would straighten out the family budget; it didn't. The ads said it would help educate the kids; it hasn't. Certainly it would do my tax returns; it can't. At least it would teach me how to prepare those fancy programs for computers; it didn't. Indeed, if I hadn't been assigned to work with my computer, it would be gathering dust in my attic.

By the end of his article, Lynch suggested that the "technically initiated" might like computers because they can make them work: "Experts say that people like me have neither the technical training nor technical inclination to make a home computer strut its stuff."

Over the past year and a half I have been conducting an ethnographic investigation of the cultures and subcultures around computation, looking at the relationships that people form with computers and with each other in the social worlds that grow up around the machines. Among the subcultures I have studied is one that has grown up around personal computation, by which I mean having your own computer in your own home. My study of personal computation began in 1978 with a questionnaire survey answered by 95 New England computer hobbyists (drawn from the roster of a home computer club and subscription list of a personal computer magazine) and continued during

1978 and 1979 with nearly 300 hours of conversation with 50 individuals who owned home computers.(1) Twenty-seven of these were drawn from the original group of respondents and the rest drawn from other sources -- for geographical distribution (particularly to represent both the West and the East coast hobbyist "cultures") and to tap the personal computer "individualist," men and women who own personal computers but who don't like to attend meetings and who are not regular subscribers to personal computer magazines. The computer hobby world, like other "technical" hobby cultures -- such as that around amateur radio and model railroading -- has a predominantly male population. This is reflected in my study. Four women responded to my survey. I met with two of them and with three other women hobbyists in the interview phase of my study.

In the course of this work I have spoken with many people whose home computers do not collect dust in their attics even though they are not paid to persevere, and I find that the answer to the question implicit in Lynch's article ("What the hell are people doing with this thing?") is not simple.

While it is true that most of the owners of home computers do have technical expertise and/or inclinations, my study indicates that what draws them to the computer is not primarily what work it can do.(2) For although there is much talk among hobbyists about making "stuff" for the computer to "strut" (devices to dim the lights and control the thermostats are among the most common gadgets written about in personal computer

magazines) much of their energy and sense of engagement is found in non-instrumental uses of the technology. In my questionnaire I asked: "What first attracted you to computers?" More than half the respondents gave reasons that were highly subjective. Twenty-six percent said that they were first attracted to computers by an appeal that was intellectual, aesthetic, or involved with the fun of "cognitive-play." They wrote of "puzzle-solving," of "the elegance of using computer techniques to handle problems," of the "beauty of understanding a system at many levels of complexity." They described what they did with their home computers with metaphors like "mind stretching" and "using the software to understand my wetware." Another 26% wrote of reasons for getting involved that seemed more emotional than intellectual. They wrote of the "ego-boost" or "sense of power" that comes from knowing how to run a computer, of the "prestige of being a pioneer in a developing field," of the "feeling of control when I work in a safe environment of my own creation."

The hobbyists who responded to my survey seemed familiar with Lynch's brand of skepticism, with people who ask them what they do with their computers and who won't take "cognitive play" for an answer. David, a nineteen year old undergraduate at a small engineering school, put it this way: "People come over and see my computer and they look at it, then they look at me, then they ask me what useful thing I do with it, i.e., does it wash floors, clean laundry or do my income tax -- when I respond no, they lose interest." David said that when he started out, he was attracted to the computer because

"I liked the idea of making a pile of hardware and software do something useful, like doing real time data processing with an amateur radio teletype," but in his list of things he does with his computer, instrumental uses are most notable for their absence:

> Conway's GAME OF LIFE in assembly code was a challenge, forced me to "think logically" and gave the pleasure of making something work the way I wanted it to... having control from the bottom level of program for that game made me feel comfortable, safe, sort of at home.(3)

David is not alone. Thirteen per-cent of my sample told a similar story. Like David they began their relationship with computation for instrumental "reasons" (a job to do, a specific task) but they became absorbed by the "holding power" of something else. A full two thirds of my survey sample either began with or (like David) ended up with a primary interest in what I shall call the "subjective computer," the computer as a material for thinking and for feeling. Clearly, to understand what people are doing with their home computers we must go beyond the "performance criteria" of Lynch's article.

In previous work I have written of the computer as Rorschach, that is, of the computer's extraordinary capacities as a projective screen for other concerns.(4) In the Rorschach the individual is presented with an ambiguous stimulus. Part of the computer's power as projective comes from its ambiguous status. In many ways it is an object "betwixt and between." A machine that touches on a sphere -- intelligence -- that man has always considered uniquely his. An object without intentionality that has been accorded enough autonomy to make "blaming the computer" a commonplace of daily life.

And part of the computer's power as projective comes from its irreducibility. As in the Rorschach whose inkblots suggest many shapes but commit themselves to none, the computer is difficult to capture by simple descriptions. We can say it is made of electrical circuits, but it doesn't have to be. A computer can be made (and several -- for fun -- have been made) of tinkertoys, and quite serious computers have been made using fluidic rather than electrical circuits. Although airplanes come in all shapes and can be described in all sorts of ways there is no conceptual problem in stating their essential function: they fly. Thre is no equally elegant compelling, or satisfying way of defining the computer. Of course, one could say that it computes, that it executes programs. But this definition has an unsatisfying element of circularity. The execution of a program can be described on many levels: in terms of electronic events, machine level instructions, high level language instructions, or through a structured diagram that represents the functioning of the program as a flow through a complex information system. There are no necessary one to one relationships between the elements on these levels of descriptions, a feature of computation which has led philosophers of mind to see the computer's hardware/software interplay as highly evocative of the irreducible realtionship bewteen brain and mind.(5) Sometimes people like to refuse the irreducibility of computation by asserting that no matter how complex the computational "product," a move in a chess game for example, "all the computer does is add." Of course in a certain sense this is correct. But saying the the computer "decided to move the Queen by adding" is a little bit like saying that Picasso "created Guernica by painting," by "making brushstrokes." Just as in theoretical

psychology there is a tension between the gestalt and the atomic, so too in computation there is a tension between the local simplicity of the individual statements that comprise a program and what one might call the global complexity that can emerge when it is run.

Tension between a local simplicity and a global complexity in the working of the computer, the elusiveness of computational process, and of simple descriptions of the computer's essential nature, all contribute to making the computer an exemplary "constructed object," a cultural object which different people can apprehend with very different descriptions and invest with very different attributes. In views of the computer's internal process, individuals project their models of mind. In descriptions of the computer's powers, people express feelings about their own intellectual, social, and political power or their lack.

Looking at the computer as Rorschach, as a projective, puts the emphasis on aspects of the individual -- from cognitive style to personal fears -- that are revealed through behavior with the computer. But of course, the computer is more than a Rorschach. The Rorschach inkblots are evocative, revealing, but they stay on the page. They do not enter the life of the individual. The computer does. It is a constructive as well as a projective medium. Many of the people I met in my study use computers metaphorically and symbolically. Their relationships with their computers enter into their ways of thinking about other things -- about politics, education, about themselves. This essay is a window on the world of personal computation, most particularly onto the ways

in which it serves as a medium for the construction of personal meanings.

My survey and interviews with hobbyists suggest that although there is much overlap, the issues that are most involved with the "subjective computer" cluster into five major groupings. First, computation is used to work through a set of issues related to personal control and mastery. Second, there are issues relating to computation and identity, most frequently, using a relationship with personal computation to enhance self image. Third, computation is used as a way to work through issues of safety and transparency, often by using the computer to construct a slice of the world that is seen as understandable rather than obscure. Fourth, computation and the articulation of political ideology. Although working through issues of control, identity, and the construction of "secure" and understandable environments have implications for individuals' relationships with politics, here that link is more direct: a relationship with computation is used as a metaphor to think through ideas about an ideal social order. Finally, there is the issue of computation and alienation, trying to use a relationship with computation to restore a sense of wholeness that may be denied in ones work life.

Control

There is a compelling tension between local simplicity and global complexity in the working of the computer and in the appreciation of a computer program. Locally, each step in a program is easy to understand; its effects are well defined. But the evolution of the

global pattern is often not graspable. You are dealing with a system that surprises. This play between simplicity and complexity is among the many things that make programming a powerful medium for working through issues related to personal control.

Depending on how the programmer brings the computer's local simplicity and global complexity into focus, he will have a particular experience of the machine as controlled or baffling, even as controlling. Both levels are there, people display different patterns of selective attention to each of them. People have different levels of tolerance for temporary losses of control and end up with different relationships to control and power in their programming work. We see a first style in Michael, an ex-programmer, now a university professor, who describes himself as "having been a computer hacker."(6) Michael was not in my sample of hobbyists. He has a terminal at home which links to a large time-sharing system, but when asked about home computers, he winced in distaste and said that he "wouldn't touch the stuff. It's too simple." His case is a contrasting backdrop for what I found to be a prevalent "hobbyist" style.

Michael described his longtime fantasy that he could walk up to any program, however complex, and "fix it, bend it to my will." As he described his intervention, he imitated the kind of hand gestures that a stage magician makes towards the hat before he pulls out the rabbit. Wizards use spells, a powerful kind of local magic. Michael's magic was local too. He described his "hacker's" approach to any problem as a search for the

"quick and dirty fix." For Michael, what was most thrilling about the experience of programming was related to using the program's flexibility (for him, defined as the possiblity of making a local fix) in a struggle to keep the whole under control.

Michael's involvement was in a struggle with the program's complexity -- what was most gripping for him was being on the edge between winning and losing. He described his experience of programming as walking a narrow line: make a local fix, stay aware of its potential to provoke unpredicted change or crash the system, test each system's flexibility to the limit. For Michael, the narrow line has "holding power." For him, weekends at the terminal with little to eat and little or no rest were frequent as was the experience of not being able to leave the terminal while debugging a program, even when the obvious need was for sleep and looking at the whole in the morning instead of trying to "fix it" by looking at it line by line all night. For Michael, the urgency of these encounters was tied to his sense that in them he was grappling with a computational essence -- the struggle to exert control over global complexity by mastery of local simplicity. The mechanism embodied in the lines of code under his immediate scrutiny is always simple, determined, certain -- but the whole constantly strains to escape the limit of his ability to "think of it all at once," to see the implications of his actions on the larger system. And this is precisely what he finds so exciting.

A second programmer, Bob, is also a computer professional, a microprocessor

engineer who works all day on the development of hardware for a large industrial data system. He has recently built a small computer system for his home and devotes much of his leisure time to programming it. Whereas for Michael, the excitement of programming is that of a high risk venture, Bob likes it as a chance to be in complete control. Although Bob works all day with computers, his building and programming them at home is not more of the same. He experiences his relationship to the computer as completely different in the two settings. At work he describes himself as part of a process that he cannot see and over which he feels no mastery or ownership: "At work what I do is part of a big system; like they say, I'm a cog." At home he works on well-defined projects of his own choosing, projects whose beginning, middle and end are all under his control. He describes the home projects as a kind of compensation for the alienation of his job. He observes that he works most intensively on his home system when his tasks at work most strongly give him the feeling that "somebody else parcelled things out..." and when he feels furthest away from any understanding of "how the whole thing fits together."

Michael and Bob have very different senses about what is most satisfying about programming. These translate into different choices of projects, into different choices of programming language and level to program at, and ultimately into what we might call different computational values and aesthetics. Michael likes to work on large, "almost out of control" projects, Bob likes to work on very precisely defined ones. Michael finds documentation a burdensome and unwelcome constraint. Bob enjoys documentation, he

likes to have a clear, unambiguous record of what he has mastered. Much of his sense of power over the program derives from its precise specifications and from his attempts to continually enlarge the sphere of the program's local simplicity.

Bob has programmed in five "high level" computer languages (FORTRAN, COBOL, PASCAL, APL, and BASIC) and has a home system which allows him to use BASIC and PASCAL. But when he works at home and can do what he pleases, Bob prefers to write in assembly language. He does not justify this preference in instrumental terms (he does not speak of speed or effectiveness), but in frankly subjective ones. For Bob, assembler means the possibility of building an environment in which he feels safe and in control.

> I prefer working in assembler even though its more cumbersome. It gives me a more direct line into what is happening in the computer. When I'm in assembler, I'm in control. I don't really feel at ease if I don't have the source code. It means that I am too much out of touch with what is going on in the machine. I do everything in assembler -- I really like the feeling of proving to myself that I can optimize better that any dumb compiler.

Like Bob, other hobbyists have built their computers from kits and many continue to work as close to the machine as possible, preferring assembly language to higher level language and in many cases even preferring to write their own assemblers rather than using commercially available ones. Two thirds of the hobbyists in my survey were like Bob in their preference for using assembly language at home and 80% of those who preferred assembler were again like Bob in that they justified this choice in affective

terms. Whereas "higher level language types" spoke about the cognitive pleasures of "problem solving" and "puzzles," those who were committed to assembler had a more emotionally charged language for talking about their work with the computer.(7) And the feelings they expressed were polarized around the issue of control: "It's a God-like feeling of creating my own universe with its own physical laws. Here I can be safe!"

Hobbyist programmers seem to mean different things when they say that machine language programming "puts them in control." For some, the reference seems objective: they are talking about what they can make the machine do. Given the primitive higher level languages available on today's personal computers, machine level programming seems to them the best instrumental solution. For others, the issue is more subjective. Many hobbyists who said they felt uneasy using systems programs for which they didn't have the source code also admitted to never looking at the listings they felt they had to have. Having access to the code was symbolic. In many cases it was apparent that machine language programming was valued not for control over a programming product, but because the act of doing it was pleasing in itself. It meant that the programmer was writing instructions that act directly on the machine, not "building on top of" somebody elses interpreter. Many hobbyists are programmers whose relationships with computation at work means sharing the machine with countless and nameless others. In personal computation they see a chance to be alone. Having the source code is a way of achieving independence. And some of the power behind the idea of working in machine language may

be the image of working on a machine over which you have full possession. It comes to you virgin. Finally, there is the issue of asserting control over an inferior. Bob, like many of the other hobbyists I spoke with, is a middle level worker in the computer industry. He does not feel very good about the importance of his job. "Proving that I am better than any dumb compiler," may make him feel more important. At work, he is an inferior position in relation not only to other people but to the machine. His time on it is scheduled to the minute. At home he is in charge.

Identity

Today's hobbyists may buy computers for what they think they can do with them, but learning about the computer, about how it works and about how to program it, usually becomes much more important than what it is being programmed for. In achieving a sense of mastery of the computer, in learning to program, in learning about the computer's "innards," people are learning to see themselves differently. Among other things, they are learning to see themselves as "the kind of people who can do science and math." This was most striking for hobbyists who had once gotten "scared out of science."

Barry is twenty eight years old, an electronics technician at a large research laboratory. He went to college for two years, hoping to be an engineer, then dropped out and went to technical school. He has always loved to tinker with machines and build things. His current job is to calibrate and repair instruments and he is very happy with it

because he gets a chance "to work on a lot of different equipment." But he came to his job with a feeling of having failed, of not being "analytic," "theoretical," of not being capable of "what is really important in science."

> Ever since I was a child I always had an interest in science, but I never had the opportunity or the passion to go back and finish college and get a real degree in science. I don't think I have a theoretical mind. I got all D's in mathematics. I have a more practical mind.

Five years ago, Barry bought a programmable calculator and started "fooling around with it and with numbers the way I have never been able to fool around before." To hear him tell it, numbers stopped being theoretical, they became practical and playful and "It seemed natural to start working with computers as soon as I could." When the calculator and the computer made numbers seem concrete, the numbers became "like him" and Barry felt an access to a kind of thinking that he had always felt "constitutionally" shut out of:

> I guess I became interested in all of this (with a sweeping hand gesture towards the computer) about two years ago: I certainly wasn't aware in what direction it was going to lead me. I'm at a point now, when I look back on history, I've seen all kinds of changes occur in me. I know these changes would not have occurred if I had not gotten involved...I'm able to do an analytical type of thinking that I never could do before. I always had a great deal of difficulty with mathematics in college which is why I never became an engineer. I just could not seem to discipline my mind enough to break mathematics down to its component parts, put it together and do it. I think if I had a chance to do it over again, and had the tool at my disposal, I think I could use it and become more. Not that I have the passion or the desire to be an engineer now. But when I was a kid, that was always the answer to what kind of job I wanted to have.

Barry claims to have "grown out of" his aspiration to be an engineer. He says

he doesn't keep engineering as a pipe dream or think of his computer skills as something that could make it real. In terms of his career plans, nothing has changed. But a lot has changed. Barry has always thought of himself as a bundle of aptitudes and ineptitudes that define him as the kind of person who can do certain things and cannot do others. Working with the computer has made him reconsider his categories:

> I couldn't do the hard math and so I ended up, what I think I am is basically a technician, a practical-type, skilled worker...but I'm doing mathematics now that I couldn't do in high school, statistical and analytical... I'll pick up the calculator, and if I don't know how to do it I'll play with that calculator a few minutes and figure it out. It's not so much that the calculator does a particular calculation, but you do so many, have so much contact with the numbers and the results and how it all comes out that you start to see things differently. Now, I'm really getting this problem area of mathematics under control and I can see what I need to learn...And I can see that I am going to get it out of the computer...
>
> I really couldn't tell you what sort of thing I'm going to be doing with the computer in six months. It used to be that I could tell you exactly what I'd be thinking about in six months. But the thing with this, the computer, is that the deeper you get into it there's no way an individual can say what he'll be doing in six months, what I'm going to be doing. But I honestly feel that it's gonna be great. And that's one hell of a thing.

For Barry, the world has always been divided between the people who think they know what they'll be thinking in six months and those who don't. And in his mind, his home computer has gotten him across the line and "that's one hell of a thing." For Barry, part of what it means to have crossed the line is to start to call the line into question. He has begun to think that people might be different if learning were different. When he was in school, his inability to do the kind of mathematics he had "respect" for turned him off to learning. The computer put learning in a concrete form that he could

participate in: "When I write in assembler I feel that mathematics is in my hands." He has three children, has bought them their own calculators, and encourages them to "mess around with the computer." ("I know they are going through the same problems I had and I would like them to have a better start than I did.") For Barry, the computer represents a better start, not because it will teach his children a particular subject but because it taught him "not to be afraid of learning."

Personal computers are certainly not the only hobby that people use to enhance their sense of identity. For the vast majority of those surveyed, "hobbies" have always been a way of life. Almost ninety percent of them had been or were presently involved in a hobby other than computation, most usually in another "technical" hobby, such as photography, ham radio, or model railroading. Fifteen per-cent of the hobbyists surveyed were using their computers to "augment" their participation in another hobby, for example, using it to keep an inventory on motorcycle parts, figure out ideal compression ratios for racing cars, interface with amateur radio equipment. For thirty-one percent of them, a computer at home had replaced another hobby. People spoke of these abandonned hobbies as "good experiences" that had increased their confidence in their ability to think through problems and bring projects to completion. But in our day and time there are several ways in which a computer hobby can be special. People spoke about their "switch to the computer" as making them part of something that was growing and that the society at large "really cared about." Gregory is in his mid-forties, has worked as a salesman in the

electronics industry for all of his working life. For two years, his computer shared space in his study with an elaborate model railroad system. A year and a half before I met him he had bought a new hard copy printer and a graphics plotter. In the overcrowding that followed, the trains had finally found their way to storage in the basement.

> Nobody ever really paid attention to my model railroad stuff, although a lot of the circuitry that I did there was just as complex as what I'm doing now with my computer...but people would look at it and they would say "that's cute." The computer is my own thing, but it's part of the real world too. And if my kid becomes good at it, it will mean something.

Different people have different senses of what the computer is allowing them to become a part of. For some, like Gregory, having a computer and "getting good at it" means crossing a frontier that separates "tinkering" from "real technology." The world sees their computer hobby as serious (several commented that friends and neighbors hardly even look at it as a "hobby," as though the word were reserved for frivolities) and they start to see themselves that way too. These people may have technical educations, but they often feel that they have never been part of what was most exciting and important in the scientific and technical cultures. Working with computers, even small computers, feels technologically "avant garde." A much smaller group of hobbyists (but a group whose numbers can be expected to grow in the years ahead as personal computers become more accessible to the non specialist) have always felt completely left out of the scientific and technical worlds. For them, owning a computer means crossing the "Two Cultures" divide.

Alan, a twenty-nine year old high-school French teacher who describes himself

as having "a love affair with a TRS-80" has always felt he "wasn't smart enough to do science."

> After Sputnik, when I was in grade school and then in Junior High, there was all that fuss, all the kids who were good in math got to be in special classes. Rockets were going up...men trying to go to the moon. Decisions about things. Scientists seemed to be in charge of all that.

Alan majored in French ("It was easy for me...my mother is from Montreal.") and took up carpentry as a hobby. And although he was good at it, it only reinforced his sense of not being able to do intellectual things, which in his mind meant not being able to "do anything technical." If when he bought his calculator, Barry crossed a line which meant to him that he had become a person who could expect change and excitement in his intellectual life, for Alan, the TRS-80 helped him cross a line to become a member of a different culture, a culture of "powerful people."

For Barry, Alan, Gregory, relationships with computation have enhanced self image. There is another way in which working with a computer can influence an individuals' sense of identity. This is by providing metaphors for thinking about oneself. I met many hobbyists who were fascinated by the idea of someday being able to trace out the complex relationships of electronic events, machine and assembly language instructions, and higher level language commands within the computer. The image of these many levels of intelligence built one on top of another provoked reflections on how people might work, about how they might be like machines or not like machines. Some of this informal

epistemology was implicit, for example in comments about "using the computer's software to think about my wetware." And sometimes, although less frequently, the issue became quite explicit. Conversations that began in discussions of household "robotics" projects, like a plan to build an energy monitoring system led into discussions of how someday it would be necessary to build programs that could better represent the system's knowledge to itself, and from there into epistemological reveries: Were these the kinds of self representing programs that ran inside of peoples' heads? Do people have different kinds of self representation programs for representing different kinds of knowledge, like the knowledge of a dream and the knowledge of being awake. What kind of self-representation program could allow us to think about and then forget our dreams?

Ideas about computers, about how they work, about what they can and cannot do, were used to assert and sometimes to elaborate ideas about people. In some cases the computer experience left people feeling that men and machines are both rule driven, that people work on programs, that intelligence is "more and more complexity, all piled up." Hobbyists are very limited in the amounts of memory that they have in their computers, and "having infinite amounts of memory" at ones disposal was a recurrent image for what would be needed to make a computer that would be a true artificial intelligence. But many of the hobbyists with whom I spoke had a very different reaction to the prospect of machine intelligence and the question of the relationship between man and mechanism. Their brush with computation led them to reflections on the "ineffable" in people, or as

one put it: "You can't put a spark of life into a computer, you can have all these programs talking to each other, but you told them to do it. In the end you can't have a spark of life...That spark of life, well, that must be God."

Building Safe Worlds

Descriptions of what it is like to work with your own computer frequently used the world "safe." People talked about feeling safe and secure in the world they had built with their home computers, a world where there were few surprises and things didn't change unless you wanted them to. Of course, there was much talk of problems, of false starts, of frustrations. There are "bugs" in hardware and in programs. Things don't work; things go wrong. But bugs, with time, become "known" bugs. For Joe, an insurance salesman in a small North California suburb who owns a second hand Commodore PET "with a lot of hardware problems," they become almost like "friends": "And then you turn the machine on, and you systematically check for your 'old friends,' and I swear, finding them there has a certain reassuring element."

Fred sells electronics components for a large electronic supply house. He narrowly escaped starvation in a prisoner of war camp during World War II, and from that experience he says that he took "a sense of optimism." "I mean, if there is something out there and you want to do it... Do it, understand it, act." Fred has tried to live that way. He is active in local politics; he keeps up with the news; he writes letters to the editor

of his town newspaper. He bought his TRS-80 on an impulse because "it didn't seem that you would be able to understand American society any more without being involved with computers. When it comes to working with his computer, he wants to know "how things work."

> "I live in an economy, and I don't understand how things are happening. I watch the energy crisis; I don't understand why it is happening that way. I drive a car, and I don't really understand how the car works. I have this beautiful couch, and I don't really know how the cloth on it is made. People used to understand more about how things worked. We live in a world where we don't know anything about anything. I don't want that to happen with the computer. I want to know exactly how things work. If not, I'm going to pass down this confused feeling to my children. And they will be afraid of the world.

When hobbyists like Fred spoke about "wanting to know exactly how things work" they were usually talking about a relationship with computation where they could think in terms of understanding "everything." The desire for this very particular kind of understanding was often framed in terms of wanting to know how a system is built up from level to level. Fred for example, expressed sharp frustration at the gaps in his ability to follow the system through.

> There is a big gap in my own mind between the fact that an electrical circuit can be on or off and the binary number system...and again from there to the BASIC language. I've got to understand all of that. I'm trying to narrow this thing down so I can follow the continuum which I can't at this moment. I'm going to the users group meetings and talking to people and reading books and some of it is helping but I am really frustrated. I want to be able to follow through.

Fred felt that in his life too many things were getting out of control. His desire to

understand the system from the and/or gates through the flip flops, the machine language, the assembler and up, seemed to express his political frustrations. Many of the hobbyists I spoke with were like Fred : their desire to understand the computer "completely" seemed associated with frustrations at how incomplete was their knowledge of other things. People wanted their computers to have a transparency that other things in their life do not.

Politics

Fred's relationship with his computer, a computer whose documentation has become as important as the computer itself ("I don't want to lose track of what I'm doing...I want to be able to see the whole thing in my mind.") is heavily invested with a desire for a kind of personal control and a personal relationship with knowledge that can be passed on to his children. Although advertisements for personal computers have stressed that they are an investment in your child's education -- that computers have programs that can teach algebras, physics, the conjugation of irregular French verbs -- Fred, like other hobbyists I spoke with, don't talk about the importance of giving their children a competitive advantage in French, but of a competitive advantage in "the computer." Most hobbyists feel that the stakes are high. They believe that computers will change politics, economics, and everyday life in the 21st century. Owning a piece of it, and having complete technical mastery over a piece of it is owning a little bit of control over the future.

Larry, a member of Fred's computer club is finding that his computer offers him a way to challenge the school's judgments of his child's abilities. A year before I met him Larry had bought an Apple computer for small business use and ended up bringing it home so he could spend more time programming and let his kids play with it. His twelve year old son Joe, had been judged "backward" by his teachers through six years of schooling. His math scores were low; he didn't read. But Joe picked up the manual for the Apple, taught himself how to use the game packages, and then taught himself how to program in BASIC in order to write his own games. In three weeks Joe was writing games that demanded the use of equations and a knowledge of geometry that his teachers claimed he didn't have. Larry feels that he has been intimidated by the school's evaluation of his son: "All these years they told me he was backward, so I believed he was backward. Now I think that he just wasn't interested And then, maybe everybody treated him like he was stupid. I am not going to sit still for it any more."

Larry is starting to demand more from his son's teachers. He shares with Fred an optimism about what computers will mean for politics because "people will get used to understanding things, of being in control of things and they will demand more." Other hobbyists I spoke with share this optimism, but what they mostly have in common is a style of talking about computers and politics. Their conversation on the subject is not about what happens when computers are used by government agencies, or about privacy, or about simulations for planning. Their associations are more metaphorical. For them, the

computer provides a place to play with a kind of understanding that they don't feel is possible in other areas of life. Images of computational transparency and of "knowing how it works" were associated with a kind of politics where relations of power could be transparent, where work would facilitate a rich and balanced cognitive life, and where decentralized power would follow from decentralized information resources.

For many hobbyists with whom I spoke, the relationship with their home computer carries longings for a better and simpler life in a more transparent society. CoEvolution Quarterly, Mother Earth News, Runner's World, and Byte magazine lie together on hobbyists' coffee tables. Small computers become the focus of hopes of building cottage industries that will allow the hobbyist to work out of his home, have more personal autonomy, not have to punch a time card, and be able to spend more time with his family and out of doors.

Some see personal computers as a next step in the ecology movement: decentralized technology will mean less waste. Some see personal computers as a way for individuals to assert greater control over their children's educations, believing that computerized curricula will soon offer children better educations at home than can be offered in today's schools. Some see personal computers as a path to a new populism: personal computer networks will allow citizens to band together to send mail, run decentralized schools, information resources, and local governments.

Many of the computer hobbyists I have interviewed talk about the computers in their livingrooms as windows onto a future where relationships with technology will be more direct, where people will understand how things work, and where dependence on big government, big corporations, and big machines will end. And they represent the politics of this computer-rich future by generalizing from their special relationships to the technology, a relationship characterized by simplicity and a sense of control.

Alientation

A final issue, closely related to personal computation as political metaphor, is the issue of computation and alienation. Most people express only small parts of themselves in their work which is often repetitive and in which they may feel that they function as machines. This very widespread and very unhappy work situation is shared by the computer programmer who may experience it in a particularly sharp form. The programmer is typically in a situation where he or she is in touch with only a very small part of the problem that is being worked on. Increasingly, programmers work in large teams where each individual may have very little sense of the whole, of how it all fits together.(8) And an operating system written by someone else, perhaps so complicated that it is not fully understood by any one person at all, stands between the programmer and the machine. Hannah worked as a programming consultant for large business systems for ten years before starting her own consulting company through which she free lances her services to

other computer hobbyists. To her, nothing was more depressing than working on a tiny piece of a problem, often "not even knowing what the whole problem was." For Hannah, concerned about conservation, health, ecology, there was something "unhealthy" about the "lack of balance" in her mental life. She liked working with computers at home because she has more control of her life: she can work on it with her family, she can do it when she wants. But she says that what is most important is that "I can finally think about a whole problem. I used to feel that I was party of somebody else's computer program." Many other hobbyists I spoke with shared Hannah's experience as programmers or as "team" engineers and shared her concern with understanding "wholes." The experience of programming in such a work group may sensitize people to the problems of cognitive alienation. The image of "balance" came up often. Karl, for example, a hardware engineer in a microprocessor firm, had a lot to say about his mental ecology.

> In the first half of an engineering project there is literally nothing coming together, and that's especially when I find that I need to go home and put something together; make cut glass lamps and so on, like the one on the living room table...But then towards the middle of the project, things start to pull together on the job. I lose the need for cutting glass and I just want to come home and watch TV.
>
> If you never get to finish things at work, if your job is basically making little pieces and its somebody elses job to fit them into a whole, then working with the computer at home should be in the service of getting it all together: doing the whole build up from machine code to finished product. That would make you feel in balance.
>
> You know, when I was working on my masters thesis, I was also a custodian for a seven room apartment building, a seven suite building. And, it was very nice that we came home from the library and had to hammer nails into boards.

When I spoke to Karl he was at a point at work where "everything seems pretty disconnected." That is, if he didn't have his computer, it would have been "glass cutting time." Karl was thinking about work with his home computer as a corrective to fragmentation on the job. He was trying to get some complex and interconnected hardware working on his home system.

> I'm starting out with a bunch of microprocessors replacing simple interface hardware for keyboards and an intelligent keyboard and separately intelligent CRT display and third processor doing computation, a fourth processor tying everything together and that type of thing...At work now I absolutely can't tell what belongs with that, so I guess that's the reason I've been mapping out hardware for multiprocessor chips when in fact I could probably get away with five dollars worth of chips and keep doing what I'm doing. So, I guess you would say that my choice of projects is not always rational.

For Karl, using hobbies for "balance" meant a sometimes "irrational" choice of projects but he seemed confident about his ability to understand and organize a whole system. For other people I interviewed, having the chance to try out "whole projects" at home, having the chance to work on complex problems on their home systems, was a chance to test capabilities of which they were less confident and about which their jobs gave them no good feelings: "With my computer at home I do everything...I see my whole self, all of my kinds of thinking in the programs I write...I never got to see all of my kinds of thinking on the job." For them, having a computer at home meant thinking experiences where they could learn what their capacities were, where they had a chance to try things out. Programmers have watched (for those who are too young, the story of the process remains alive in the collective mythology of the shop) their opportunities to exercise their

skill as a whole activity being taken away. They have watched their work being routinized, being parcelled out into the well defined modules that make up the tasks of the structured programming team. This lived experience at work make programmers particularly sensitive to the parcellization of knowledge and to the alienation from a sense of wholeness in work. And they bring this sensitivity to their new hobby. Whether or not they consciously had it in mind when they bought their home computers, their relationships with computation at home often suggests a search for a lost relationship with what is exciting about the computer and for a lost sense of control over a whole process. Most people do not work in jobs that have as clearly articulated (or as recent) a myth of a "golden age." But most people feel that their work does not allow them full expression and that there was a time when things were different. It is easy to identify with the programmer/hobbyist: one feels in touch with people living a particularly charged reationship to the experience of job routinization and who have a powerful material in hand for concretizing fantasies of how things might have been different and how they might be different once again.

* * *

In this essay I have bracketed the question of how the world might change because of instrumental uses of computation in order to focus on how people are changing because of essentially non-instrumental uses of computation.

In studying the hobbyist experience I have found people, largely people with technical backgrounds, in intense involvements with machines. Not surprisingly, given the low proportion of women in the technical professions, most of them were men. Most of the people I met have long histories with hobbies involving other technical objects: with cameras, radios, with model building of all sorts. And many, having passed through other hobbies, came to see their relationship with the computer as being special. Their relationships with computation became involved with their feelings about politics, education, and about themselves. They describe their work, or rather their leisure with the computer as different than what they have done before with other hobbies. They describe it as an involvement with greater personal consequence.

Some of that sense of consequence comes from an historical moment: the hobby is seen as signifying a place in the "avant garde." Although in some circles to be called a "computer person" is to be addressed with a term of derision, the hobbyist tends to experience his identification with the computer with pride. And some of the sense of consequence comes from the fact that many hobbyists are using their hobby to experience an individualistic and independent relationship with computation that is mythologized as belonging to a now-past golden age of the programmer. But most of the sense of consequence comes from the holding power and intensity of the time spent with the computer. What is there about these people and these machines that makes possible

relationships of such power and complexity?

For me, the relationships that hobbyists form with their home computers can be partially captured with a metaphor of the "mind" and the "body" of the machine. The "mind" of the computer is that side of computation that involves thinking in terms of high level programs. In this metaphor, relating to the body of the computer means not only working on hardware, but also, and indeed especially, working with programs in a way that is as close as possible to the machine code, that is to say as close as possible to the core of the computer, its Central Processing Unit. In terms of this metaphor I have found that the prototypical hobbyist is trying to get into a relationship with the body (rather than the mind) of the machine, seeking to assert power and control in the relationship with the computer, and to create safe worlds of transparent understanding. In trying to find concepts for thinking more clearly about what draws the hobbyist to this kind of relationship with the CPU and about what its meaning might be, I find three issues particularly salient. I think moreover, that although I formulate them here in terms of computers and people, these issues open out to more general understanding of the subjective side of other technologies as well.

The first issue goes back to the question of control. The hobbyist complains of a work situation where he or she suffers from the constant presence of intermediaries. Bureaucracies stand between the hobbyist and the computer, a bureaucracy that schedules

the computer, that decides its up and down time, that apportions the work for its software design and decides on priorities and on procedures for access to it. At work, when something goes wrong with the system, it is usually the fault of an intermediary person, one of the many "somebody elses" who deals with the machine. Or it may be the fault of a technical intermediary: one of the many elements in the computer system that mediate between the user and the bare machine: a compiler, an interpreter, an operating system, someone else's program. At home when the hobbyists work directly with the Central Processing Unit they are all alone with the computer, in complete and direct control of the machine's power. And when something does blow up, the situation at home is simple because it is just between them and the bare machine. They can hunt for their own bugs.

When a FORTRAN program is compiled and run, the events in the machine are far from being in one to one correspondence with the steps of code written by the programmer. The frustrating sense of always being in an indirect relationship to what one is doing is further exacerbated when the compiled code is run by an operating system which allocates memory, mediates control of peripheral devices, and even interleaves the program with other programs. At home the hobbyist feels in immediate touch with the essential machine: nothing and noone stands in an intermediate role. This relationship is often epitomized by the possibility of following programs step by step as their instructions passs through the Central Processing Unit, something that would be impossible on the larger system. They can do this physically, by making the program run one step at a time,

or conceptually, by imagining the changes of state of the CPU, a particularly clear thought experiment for debugging.

When the program is seen locally, instruction by instruction, programmers can envision the changes in state of the whole system as being produced by specific actions of the CPU. And if they suspect that the bug is in the hardware, they can pull out an oscilloscope and see whether the CPU is doing what it should in response to a given instruction. They can figure out where the signals should be going, they can collect their own evidence for what is going wrong, trap and fix the bug themselves. Again and again in my interviewing I heard about the pleasures of debugging — of "going in with meters and scopes and tracking it down." The procedure exhilarates. With every successfully tracked bug comes an affirmation of power over the machine.

The issue of control was often explicitly recognized by the hobbyists I interviewed. But they lacked a language for naming the second issue which has to do with a notion referred to as "syntonicity" within the psychoanalytic tradition. Syntonicity implies that we should look for "body to body" identification at work in powerful relationships with technology: the body of the person and the body of the machine. This approach would try to understand the power of machine language in terms of peoples' ability to identify with what is happening inside of the machine. The CPU of the hobbyist computer lends itself to personal identification. The action of transferring what is

conceptually almost a physical object, a byte of information, from one physical place (a register) to another is very close to concrete and familiar human actions. Working in machine language means working with a model of the computer as having registers that contain certain information that needs to get moved around from place to place. It is possible to imagine the registers, to project oneself into the physical situation of moving things from one to another. The metaphor is concrete and spatial. You can imagine finding them, feeling them, doing something very simple to them, and passing on. For many of the people that I met in the hobbyist culture, getting into this kind of identification feels safe. It makes the machine feel real.

There is a third issue raised by the hobbyists' relationship to the CPU. It is an aesthetic one. The generation of hobby computers that was born in the 1970s are very primitive machines. The hobbyist thinks of much about them as "klugey," a computerists way of saying that one is dealing with a compromise, a collection of patches whose structure has been dictated by arbitrary corporate decisions, by economic necessities. The corner of the hobbyist machine that seems to them to have the greatest "intellectual integrity," that distills what they feel to be a tradition of some of the best ideas in computer science, that comes closest to being "clean," is the CPU. And so it is natural for the hobbyist to seek the closest possible contact with it. For a culture in which there is a widely shared aesthetic of simplicity, intelligibility, control, and trasparency, getting into the "guts" of the machine and working in machine code seems the best way to use the

personal computer as an artistic medium.

* * *

I would like to end with a personal reflection on what I found. As an ethnographer I use interviews and observation to enter sufficiently into people's lives to develop a sympathetic understanding of how they look at the world. And then I take this experience and try to distill from it those elements that will make the lives of the people I have been studying intelligible and meaningful to others. The hobbyists I interviewed are excited, enthusiastic, satisfied with what they are doing with their machines. It seems appropriate to report this enthusiasm and to try to capture a sense of the pleasures and satisfactions that these individuals are getting from from developing "non alienated" relationships with their computers, from "understanding" machine systems from the "bottom up," and from feeling satisfied that they finally have found models of transparency and order for thinking about the kind of world they would like to live in. But there is a darker side to the picture. Will these individual satisfactions of personal computation (which seem to derive some of their power from the fact that they are at least in part responsive to political dissatisfactions) take the individual away from collective politics? People will not change unresponsive political systems or intellectually deadening work environments by building machines that are responsive, fun, and intellectually challenging. It would certainly be inappropriate to rejoice at the holistic and humanistic relationships that personal computers offer if it turns out that when widespread, they replace religion as an opiate of the masses.

NOTES

1. I use the word "hobbyist" in this essay to denote someone who owns a personal home computer, whether or not that computer was built from a kit. The term is problematic. "Personal computerist" might be more accurate but seemed to me to be even more awkward.

2. The statistical results of my survey certainly support the idea that today's satisfied personal computer consumer has technical training, and/or inclinations. Eighty-three percent of those polled had majored in some scientific or mathematical field in college, and most of them (60%) actually made their living from some kind of work in the computer industry: building computers, selling them, servicing them, or programming them. Twenty-eight percent of the respondents had majored specifically in mathematics or computer science. Thirty-eight percent of them had done some graduate work in a scientific or technical field. For over a third of the hobbyists surveyed, when it comes to computers, they have the inclination rather than the training: 35% claim that everything they know about computers is self taught.

3. The GAME OF LIFE refers to a popular class of computer demonstrations. Their name refers to an analogy with an evolutionary image of the emergence of the complexity and variety of living organisms from very simple origins. In these demonstrations transformation rules are applied repeatedly to patterns of objects on a checkerboard. Conway's ingenuity consisted of finding classes of simple transformations that would give rise to surprising, complex, and varied effects. The popularity of the game springs from many sources: the biological referent fascinates as does the interplay between simplicity and complexity. People describe a thrill from getting genuinely surprising results from the application of simple rules. Its particular popularity among hobbyists also reflects something about the kind and level of programming required to run the game on a small computer. It is beyond the stage of routine programming, but not too far beyond. The main problem in programming the GAME OF LIFE is speed. The faster it can run, the more dramatic the effect. Having it run at an aesthetically satisfying speed on a small computer requres ingenuity. And this ingenuity has immediate, visible consequences.

4. Sherry Turkle, "Computer as Rorschach," Society/Transaction (January-February 1980).

5. See for example, Hillary Putnam, "Minds and Machines," in Alan Ross Anderson (Ed.), Minds and Machines (Englewood Cliffs, N.J.: Prentice Hall, 1964).

6. For Michael, it was his style of programming that led him to identify with what for him was a computer "subculture," that of the hacker. His process of identification seemed analogous to that of a creative independent virtuoso who recognizes his peers not by the "job" they do nor by their academic credentials, but because they share his sense of the personal importance, the urgency of creating in the medium in which they work. Many hackers have dropped out of academic programs in computer science in order to devote themselves exclusively to computers. Based neither on a formal job nor on a specific research agenda, the coherency of the hacker subculture follows from a relationship with the "subjective computer," that is, with a set of values, a computational aesthetic, and from a relationship with programming that may be characterized as devotion to it as a thing in itself. In university settings all over the country, where hackers are often the "master programmers" of large computer operating systems, academic computer scientists complain that the hackers are always "improving the system," making it more elegant according to their aesthetic, but also more difficult to use.

7. Among the hobbyists I surveyed, preferences for the level of programming language were strikingly associated with what people saw as most exciting about working with computers. For example, forty percent of those who preferred assembly language wrote about an "affective" issue as the source of their engagement. Among those who preferred to work in higher level languages on their home computers (BASIC and PASCAL) only 10% wrote about an affective issue. But nearly eighty per cent wrote about what I have called "cognitive play" (problem solving, puzzle solving.)

8. See Philip Kraft, Programmers and Managers: The Routinization of Computer Programming in the United States (New York: Springer-Verlag, 1977).