IV

Women in The High Mainframe Phase, 1960-1980

The programming profession was expanding rapidly and continuously all through this period. The middle period of computer programming was, above all else, the period of initial engagement with the external world. It was the period in which the number of computer programmers rose from a few tens of thousands to hundreds of thousands,[1] and the number of computers from the hundreds to the low millions. Programming went from an obscure, if esoteric, field of technology to an ascendant field, not yet predominant, but with the promise or threat of ultimate predominance visible.

The ascendancy of programming was based on a series of inventions which made computers far more usable than they had previously been. The computer’s logic circuits were built out of transistors rather than tubes, and transistors would soon be replaced by integrated circuits. In the main memory, ferrite cores persisted for the time being, but they had been the least inadequate components of first generation computers.

For secondary storage, magnetic disk drives became available,

---

1. At the time of ENIAC, in 1943-44, there were, depending on definitions, perhaps as few as a dozen programmers. By 1960, there were about 13,000. By 1970, there were over 250,000, more than 600,000 by 1981, and, ultimately, there would be more than a million and a half programmers by 1993.

making really large quantities of storage, a megabyte or more, available in any desired order. The disk drive’s speed of access meant that the core memory could do multiple duty, holding first one part of a large program, and then swapping it for another part, so as to yield most of the benefits of an impossibly expensive quantity of core memory. Of course this swapping could be overdone, resulting in a kind of disk-to-core traffic jam known as ’thrashing.’

Superimposed on these hardware inventions, and made possible by them, were two software inventions: the operating system and the high-level language. Taken together, they permitted a programmer to write programs with minimal reference to the details of the machine, and in a language comparatively close to his own. The operating system and high-level language also possessed at least rudimentary facilities for dividing a program up into manageable chunks and sharing these chunks between programs. Further, the operating system liberated the computer from the human slowness of its operator, and drastically multiplied the computer’s availability. The net effect of all this was to control the complexity deriving from the computer itself.

The computer became a locus of rational order, to which fragments of information were continually attaching themselves and thereby becoming orderly themselves. Once there was adequate systems software[1], bits of program began to accumulate around the

---

[1] The generic term for operating systems, language translators, etc.

systems software, and the accumulation was progressively sorted out, reorganized, consolidated, and cataloged for use in other programs. Something similar occurred with data, with miscellaneous sets of records, many of them on paper, being unified into large central electronic archives. It was this accumulation, as much as the system software per se, which transformed the nature of programming. The rise of system software had an effect analogous to the rise of human language a million years ago-- it rendered progress cumulative, and not subject to regular liquidation.

This control of complexity reduced the cost and difficulty of programming tremendously. Reducing the cost of programming made it feasible to program new applications, which were typically smaller, more diverse, more numerous, and, one might add, more individually complex. Many of the first areas to be computerized had long been subjected to routinization and standardization by way of conventional forms such as printed ledgers, invoices, etc., even if they had not used tabulating machines, the predecessors of computers. The newer areas of computer usage would not have been feasible if programming had remained expensive.

Easier methods of programming created an expansive mood, notwithstanding periodic expressions of dismay. There were limitless numbers of things which could be done with computers if they could be done cheaply enough. Computer programming assimilated new bodies of knowledge, and certain segments of computer programming came to define themselves as the art of assimilating new bodies of knowledge, as a kind of applied philosophy, in short. As Herb Grosch, then DATAMATION’s columnist?, and eventually to be president of the Association for Computing Machinery, put it in 1960:

> ...we have to wade through the artifacts of a whole discipline like chemical reactor theory or airline reservations practice, or even find ourselves forced to codify and regularize where no set discipline exists... the provision of ...[programming] tools should occupy only a few of our professionals, not dominate the working thoughts of a majority... To pioneer a new application expands our horizon far more than to redefine the use of the semicolon in UGHTRAN, and should stimulate a far larger percentage of our best people.[1]

Like Philosophy, which one philosopher called "an intellectual poaching licence," Computer programming is fundamentally an incorporative discipline.

This is in striking contrast with traditional engineering. Engineering disciplines had reached their limits of expansion when they had ceased to be within the realm of mutual comprehensibility. For the various branches of engineering to come into being, they had been obliged to define definite common bodies of knowledge, which thereafter limited their scope. In the 1840’s, mechanics, the predecessors of mechanical engineers, could agree on their desire for the status of professionals, and on the conventionally middle class virtues, but not on the

---

1. "Plus & Minus by Herb Grosch," DATAMATION, January/February 1960, p. 32

technical content of the proposed profession.[1] If they could not agree on the content of a profession, then they could not say what work was well-done or ill-done.

When mechanical engineering crystallized, it did so as a definite field of endeavor: shop and steam practice. Mechanical engineers were not millwrights-- they did not concern themselves with the full range of machinery which was in existence by the 1870’s. Instead they designed and built machine tools, and used them in machine shops to build steam engines which they also designed. Mechanical engineering more or less immediately split off from civil engineering. The civil engineer’s surveying and cost estimating skills were not relevant, and civil engineers were likely to take refuge in blanket denigration of a field they were not competent to criticize. If a civil engineer could not tell good mechanical engineering from bad, the only way he could protect his status was to take the position that all mechanical engineering was a trashy sort of civil engineering.

The mechanical engineers were obliged to operate the steam engines until such time as they could simplify the engines to the point where they could be operated by less skilled operators. When such operators became available, the mechanical engineers made it quite unmistakably clear that these people were not engineers,[2] and then they retired in lordly seclusion to perfect

---

[1] Monte A. Calvert, <u>The Mechanical Engineer in America, 1830-1910: Professional Cultures in Conflict</u>, The Johns Hopkins University Press, Baltimore, 1967, pp. 31, 39, 40

[2] Calvert, op. cit., pp. 127, 189-90

the arts of machine shop practice and steam engine design.

The mechanical engineers then shut the machine shop door and refused to take seriously all further developments in engineering. They took a dismissive attitude towards electrical engineering as a subordinate skill,[1] and as for the new mass-production engineers such as the automotive engineers, the old-line mechanical engineers dismissed them as producers of shoddy wares.[2] To do otherwise would have obliged the mechanical engineers to learn about all the artifacts of Fordism.

Computer programming was remarkably free of this sort of snobbery. The identity of computer programmers was tied up in looking outward for new things to program. These new tasks were generally forthcoming, and more and more programmers were hired to program them.

The extreme rate of expansion of computer programming meant that employers found programmers where they could, and however they could. Often they raided each other’s employees. Headhunting or the equivalent was a normal mode of recruitment. Gentlemen’s agreements to the contrary would not have survived for very long.

At the beginning level, no particular qualifications were required. There simply were not enough qualified people to go around. Employers found people where they could, and trained them to program at company expense. As late as 1986, company-financed

---

1. ibid. p. 215

2. [source this in Sinclair?]

programming training was still going on.[1] The basis of selection might be personal acquaintance, or it might be a psychological test,[2] or the employer might recruit college graduates. There was almost a presumption that anyone who was not obviously unsuitable was to be given a try.

Programming was open to the most marginal groups of outsiders, those beyond the protection of the first civil rights acts. Some examples were the severely disabled, whose political coming of age was Title 504, and prison inmates and ex-convicts, who remain under legal disabilities. Intelligence and diligence were about the only requirements to become a programmer.

To draw upon the intelligence of the disabled, the computer establishment first made minimal concessions to the special problems of these people. But over time, the concessions became greater, as the remaining restrictions came to be seen as counterproductive.

Sometime in the fifties or sixties, Ida Rhodes of the National Bureau of Standards was giving instruction in computer programming to the deaf-and-dumb and blind.[3] The deaf, dumb, and blind were the most socially acceptable of disabled persons. There was by this time an established practice of teaching them skills involving literacy, and an establishment of teachers,

---

[1] See Janet Dight, "Grow Your Own Programmers," <u>DATAMATION</u>, July 1, 1986, p. 75

[2] Such tests were of very dubious validity, as Gerald Weinberg (op. cit., pp. 153-58, 170-76) points out.

[3] Eric A. Weiss, "Biographies: Obituary of Ida Rhodes," <u>Annals of the History of Computing</u>, 1992, 14[2]:58-59

schools, Braille transcribers, etc. So this first initiative would not have meant much if it had not been a prelude to greater things.

In 1972, IBM began a program, in cooperation with the state of Virginia, to train the disabled as computer programmers. By 1986, a series of similar programs in thirty-one cities had produced more than 1900 programmers, 82% of whom had found jobs. Now yet another program was being started in Dallas, Texas, sponsored by IBM and twenty other companies. Its first class of students had gone from being government dependents to earning the prevailing entry-level salary: $21,000 to $25,000 a year. At this stage, concessions were comparatively minimal. It was expected that disabled programmers would be able to sit in wheelchairs for long periods of time, and would not require much in the way of work reorganization.[1]

About 1975, a similar program had been founded in England, organized by the British Computer Society, which had set up a 'Specialist Group for the Disabled.' Again, there was the same mixture of corporate, government, and academic sponsorship. The English program was even more activist than the American one. Whereas the American program concentrated on the comparatively fit disabled, their British colleagues were going into the rehabilitation hospitals, to find potential programmers not merely among government dependents, but rather among government wards. The organizers cheerfully designed new and better computer

---

1. Robert J. Crutchfield, "News in Perspective-- Training: Back to Living," DATAMATION, September 15, 1986, p. 48, 52, 56

terminals designed to be used by quadriplegics. One of the quadriplegics, Geoff Busby, was reported to be typing with his nose. With the new equipment, he eventually attained a typing speed of twenty words per minute. Oddly enough, the most severely disabled people were the hardest workers and the fastest learners, probably because they had been bored silly, sitting around institutions.[1]

Of course, the long-run direction of computer programming was towards personal computers, and towards quite a large measure of communication by electronic mail. There would be fewer restrictions on working at home, under such physical conditions as the programmer saw fit. The time was coming, when this reliance on electronic mail would yield a paradox. A talented programmer might be locally famous (or notorious if you prefer) not only for his expertise, but also for his forceful manner, and abrasive personality. And yet, this programmer might also be so handicapped that he could not speak in such a manner as to be readily understood by those not accustomed to him.

However, all these people were intelligent, and they could be diligent when the circumstances justified, and so they became computer programmers. What applied to those immobilized by nature also applied to those immobilized by human ordinances. Even prison inmates could take part in the computer revolution. The act of programming over-rode their status as prisoners. It also

---

[1] Nancy Foy, "International-- United Kingdom: Disabled Programmers Herald Self-Help Successes," DATAMATION, June 1977, pp. 168-CC, 168-FF

overrode seemingly crucial reservations which might have been posed concerning their moral fiber.

Programming was not, of course, the only skill taught in prison schools, but it was probably one of the most successful. By 1976, Jeff Larkin, an inmate in the Wisconsin State Reformatory who had taken his first programming class only three years before, was running his own small data processing department, with two other inmates under him, doing consulting work for various state agencies.[1] Interestingly, Larkin remarked: "This was a revolutionary idea to the people involved in the penal system-- that an inmate could do something that complicated and sophisticated." It must have been an axiom that prison inmates, the last slaves, were stupid. But this axiom was being shattered.

The fact that prison inmates were being made into programmers was a indication of just how desperately needed programmers were. Convicts are of course not bondable, that is, their honesty cannot be insured. For someone in a fiduciary position, such as a corporate manager or public official, to hire a convict for a responsible position carries a hazard. If the convict should do something amiss, or possibly even be denounced as a convict,[2] his patron would be at risk of censure.

---

[1] Jeff C. Larkin, "Letters [to the editor]: Inmate dp," DATAMATION, February 1976, p. 7-8

[2] One can imagine how an opportunistic and unscrupulous state legislator, in the style of Joseph McCarthy or William Proxmire, could have used Mr. Larkin as an expendable pawn in order to attack the governor.

Programmers do not embezzle very often, not being subject to the kinds of financial and career frustrations which afflict chief cashiers and bank branch managers.[1] Programmers' skills bring them money sufficient for their needs: indeed, their pay scale often seems to outrun their own rising expectations. Their work generally immunizes them against frustrations growing out of boredom. However, when a rare alienated programmer does decide to embezzle, and finds a way to do it, the results are apt to be devastating, involving 'deductions' from thousands or millions of accounts, use of Swiss banks, refuge in non-extraditing countries, and plans to 'disappear' with new identities. But this is of course a very rare thing to have happen.

To make a prisoner into a programmer was in effect to make him a 'Nouveau Homme,' whose past in another world was irrelevant to his present identity. Likewise, the bodies of disabled programmers were irrelevant to *their* present identities. All of them were what they willed themselves to be. Women, too, could will themselves into a new identity.

In this context, the alleged shortcomings of women were trivial. Not since the mid-nineteenth century had women's skirts taken up so much space as a wheelchair. Nor, since about the same date, had women suffered under a presumption of moral irresponsibility. Indeed, since the seraglio went out of fashion, there had been no substantial restrictions on womens' freedom to

---

[1] Much computer crime has involved cashiers and branch managers without programming skills. They had unwisely been given the ability to change account balances without proper transactions.

travel in the course of their business, at least none comparable to the "iron bars, which do not make a cage." If women brought new problems to the programming workplace, these problems were not intractable. Solutions would be devised, often by using computer technology in novel ways. It will be not be a very great surprise that women had no significant difficulty in becoming programmers, remained so in spite of motherhood, and in due course were promoted to management level.

Entrance into computer programming was probably the simplest part. There were simply not enough programmers. Formal educational programs were being set up, but that took time. As long as the number of programmers required was expanding exponentially, it was bound to outrun the supply.

The women who entered programming did so from a wide range of origins, and with a correspondingly wide range of qualifications. Some of them were highly educated in technical subjects; some were highly educated, but not in technical subjects; and some had minimal education.

Sometimes women got their chance by being secretaries in the right place at the right time. Such women had often had no previous chance to demonstrate ability at any technical or scientific work, or even to demonstrate the ability to deal in abstractions. They must have been selected on the basis of very little more than being known to be somewhat harder workers than someone from off the street would be. As Mary Lynn McCaffery of Citibank put it:

> ...I became a secretary in a department that was involved in developing time sharing programs as a new

> product for correspondent banks. I found secretarial work frustrating and I kept pestering everyone for more to do. My boss suggested I try programming, so I borrowed a FORTRAN manual and started programming and found I really enjoyed it.[1]

Nobody really gave Grace Householder a chance to become a computer programmer-- she found the chance in a power vacuum and more or less helped herself. Universities, hospitals, and kindred institutions are a bureaucratic anomaly. Most of the positions of honor and prestige are granted to distinguished scholars or practitioners, or else to self-effacing men who rarely forget that they have "no voice to speak, save as this honorable faculty may direct them." One way or another, the nominal heads of an academic or medical institution have remarkably little disposition to actually run it. The institution actually gets run in large measure by women of no great rank and no great authority: department secretaries, office managers, and suchlike. Grace Householder was an office supervisor in a small community hospital.

In 1966, Householder's hospital got a small computer (an NCR 500), and the vendor staked Householder to a minimal course of training, lasting only a week, and presumably of the "this is a computer" variety.[2] According to the practice of an ordinary business, with efficient professional management, the programming of this computer would no doubt have been handed over to a

---

[1] "Women in Management: A Conversation," (<u>DATAMATION</u>, April, 1980, pp. 131-140), p. 134

[2] eds., "'Enthusiasm is Contagious,'" <u>DATAMATION</u>, September 1977, p. 30, 34

consulting firm. Alternatively, one of the elite (physicians in this case) would have been sent off to really learn to program, on an extensive and expensive course. Or a cadre of trained programmers might have been recruited. But this was not a business; it was a hospital. Probably none of the distinguished physicians deigned to know anything about computers, and, knowing nothing, they could hardly prevent Householder from simply going into action by herself. She somehow managed to learn to program by herself, and having done so, went on to write programs to do various useful things around the hospital, and in due course to build up a data processing department.

Other women came in through the regular recruiting arrangements, especially those for college graduates. They did not have to have a background in any technical subject. Such selection as existed was often a standardized test.

Christine Millen had studied Classics at the University of London. Classics was of course the most prestigious academic subject in England, the traditional passport to the Higher Civil Service. A Classics degree did not interfere with getting a programming job. As Millen puts it, "it wasn’t necessary to have any special skills: the attitude was ’we’ll take brains and train them.’"[1] Millen did not really set out to become a computer programmer: rather, she interviewed with International Computers Limited, the British national rival to IBM. She was offered a job, accepted it without any long-term intentions, and then one

---

[1] "Women in Management," loc. cit., pp. 133-34

thing led to another.

Of course, some women did have a technical background. Women engineers were rare, but there were a few of them. Donnamaie E. White had gotten a BSEE circa 1964. She had started out doing the electrical engineering she had been trained for, but had graduated into aerospace systems design. Later, she went to graduate school, specializing largely in software for electronics design, and even did a stint as a database project manager. By 1979, she considered herself "both a hardware and software person." She was then employed in developing components to improve the performance of old computers.[1]

Women mathematicians were more common, of course. Nancy Jordan had gotten a Mathematics Degree from Wellesley, on the grounds that the employment prospects were better than Art History. As a Mathematics major, she was directed into scientific programming rather than business programming. Her first jobs were with IBM and later, Computer Sciences Corporation. Both jobs involved working on NASA contracts, starting with Gemini. Eventually, Jordan got bored with scientific programming, and put in for a job in Holland under contract to N. V. Phillips, working on an operating system. But she decided she didn't like living in Holland, so she came home again, and took a job in Computer Science Corporation's New York office, doing business programming. In due course, she moved on to Chase Manhattan, and

---

1."People: Into Engineering Early On," DATAMATION, November 1977, p. 29-30 went into management there.[1]

Thus there were many possible ways to enter programming, open to many different kinds of people. Programming was expanding rapidly, as fast as it could expand, and it needed every last one of them.

In a couple of years or so, the novice had become a more-or-less experienced programmer, possessing what amounted to journeyman status. By this time, she often wanted children. Pregnancy and motherhood were not the kind of career disaster they might have been in other occupations. Women programmers found ways to work part-time, or otherwise to keep their hands in.

This had not been the case in other occupations which employed large numbers of women without becoming feminine ghettos. Such occupations were much less forgiving than programming. If there should be weaknesses or vulnerabilities associated with pregnancy and motherhood, and hence with femininity, male colleagues and competitors were not going to go on a sympathetic strike in the name of gender fairness.

Journalism, for example, employed large numbers of women. By 1972, fully 41.4% of editors and reporters were women. By 1981, the figure was 50.1.[2] If gross numerical equivalence had been all

---

1. "Women in Management," loc. cit.,

2. Statistical Abstract of the United States, 103rd Edition, 1982-83, U. S. Department of Commerce, Table 651, "Employed Persons by Sex, Race, and Occupation: 1972 and 1981," pp. 388-89

there was to equality, women journalists would have been equal by the mid-seventies. But as late as 1988, when sexism as such was no longer acceptable, only thirteen percent of the directing editors were women. Equality at this level was not expected until the year 2055. The problem was that for a woman journalist to take time off in order to have children was commonly disastrous to her career. The top women were single or childless.[1]

Employers in journalism were often willing to compromise with working mothers, but the nature of the work meant that compromises were less satisfactory. For example, when a woman got pregnant, she was apt to be reassigned to a desk job, and in practice, this might involve a substantial demotion, to the borderline clerical level. For example, one woman who "...had been covering government, politics, and crime..." was reduced to "working on wedding stories."[2] Yet it is hard to see what other kind of desk job could have been found for the woman. Only a handful of senior editors could get to write editorials. Most of the jobs a newspaper had in its gift consisted of going out and getting facts to write about.

Tia O’Brien, a Philadelphia radio and television reporter, offers some insight into the reasons why pregnant female reporters were not allowed to go out on assignments. In 1975, she was attempting to cover Frank Rizzo, the mayor of Philadelphia,

---

[1] Mary Ellen Schoonmaker, "The Baby Bind: Can Journalists Be Mothers" (<u>Columbia Journalism Review</u>, March/April 1988, pp. 33-39), p. 39.

[2] ibid, p. 37

who was admittedly one of the more violent of American politicians. Rizzo did not want to be interviewed, and the reporters covering him were obliged to adopt paperazi tactics. In the course of these encounters, it was normal for Rizzo’s bodyguards to step on Ms. O’Brien’s feet and twist her arm.[1] The next year, she was being assaulted by the goons of Teamsters Boss Tom Magrann.[2] In short, street journalism is transitionally dangerous.

Likewise, it is in the nature of news that it comes in at all hours. A newsroom has to be manned around the clock because things happen around the clock. So it is not surprising that employers were often less than cooperative about letting a woman choose to work only days. Daily deadlines are real. The occupation does not lend itself to going off by oneself-- for whatever reason-- for weeks at a time. Pregnancy, need for a sabbatical in which to write a book, or a sudden fondness for vodka-- the reason, and its possible meritoriousness, is immaterial. What counts is the fact of absence. Another woman journalist, trying to get her old job back after maternity leave, and trying to avoid being put on nights, fought a prolonged battle with her employer, including the threat of legal action, but in the end, she had to resign her job.[3]

The structure of journalism presented a whole series of

---

1. "Tia Tells All, <u>Philadelphia Magazine</u>, August, 1991, p. 130
2. ibid, p. 193-94
3. Schoonmaker, op. cit., p. 37

constraints of time, place, and even physical force. When Mary Ellen Schoonmaker laid out her proposals for a mother-friendly newsroom,[1] there was a fundamental unreasonableness in them. She could not change the world to make things like night work, or rough-housing with political gangsters unnecessary. Instead, she was asking for exemption from the chores of journalism without penalty.

Programming was different, because it was what programmers wanted it to be. Not all programmers were women, let alone mothers, but almost all of them valued their own autonomy, and had therefore been working towards a liberating kind of programming. Most real-world constraints had been designed out. Still more constraints would be designed out in due course. The programming mother was commonly able to design a system of programming according to her needs, a system in terms of which she was not particularly unfit or handicapped.

A woman programmer who became pregnant was under no immediate obligation to quit work. Programming is no more physically strenuous or awkward than the traditionally approved knitting of infant garments, and it is considerably more lucrative. Likewise, as long as she was available to consult in an emergency,[2] the woman programmer could take a few weeks off to give birth. At

---

1. She called for: "... an adequate [maternity] leave, satisfactory child care, and being able to return to the same job or a comparable one that fits both the paper’s and the employee’s needs.", ibid, p.35

2. To "consult" in this context generally means to explain where one put something, or what one did.

this point, the programming mother had a range of choices. She could either continue to work full time (or even full time and then some) and possibly get a nanny; or she could find ways to work at home, treating programming as just another of the traditional domestic occupations of mothers.

Some women took the former approach. Christine Millen spent four weeks away from work, and returned with a promotion.[1] That was probably fairly atypical. Many mothers left work in whole or in part for much longer periods of time. There was a wide variety of options for them.

All else failing, women programmers with children could join together to organize their own software companies, in which their needs were axiomatically paramount. An example of such a company is the English firm, F International. F International was founded in 1962 by (Mrs.) Steve Shirley, to provide programming employment to women with children. By 1976, it employed 340 people, mostly married women with children dispersed over England and Scandinavia. They worked at home or in small dispersed offices, typically on a twenty-five hour week. F International specialized in exclusively custom software. The project managers were also dispersed, living near their major customers. At least at the managerial level, there was very little job rotation-- the emphasis was on cultivating

---

[1] "Women in Management," loc. cit., p. 136

ongoing relationships with customers.[1]

If F International was not formally a womens programming cooperative, then it was practically indistinguishable from one. By 1980, F International was up to 600 employees. By that time, there was a small American company run along the same lines, Heights Information Technology Service, started in 1978.[2]

Of course, these two firms were numerically insignificant, but that did not mean that they were insignificant in their influence. As Joan Wallach Scott has pointed out in her study of nineteenth century French glassworkers,[3] worker cooperatives cannot suspend the basic economic forces which act on a business. Conversely, if a cooperative functions, it is unlikely to be doing anything an ordinary firm cannot also do. Cooperatives are perhaps, a less economically destructive form of strike, and their real significance is not in the number of employees, but rather in the terms of trade and work practices which they legitimize. F International was more than anything else, a concrete demonstration that women could be usefully employed as programmers on their own terms. Because the company, or one fashioned in its likeness, was ultimately available to

---

1. "Calm View of Management," <u>DATAMATION</u>, August, 1976, p. 13-14.
This is a profile of Suzette Harold, then-managing director of F International, and an exemplary long-term employee.

2.
Edith Myers, "Home is Where the Work is", <u>DATAMATION</u>, February 1980, p. 77-79

3. <u>The glassworkers of Carmaux; French craftsmen and political action in a nineteenth-century city</u>, Cambridge, Mass., Harvard University Press, 1974, pp. 167-87

discontented women programmers elsewhere, it defined the kind of terms and conditions of employment that regular employers would offer to women and mothers.

Far more common than women's cooperatives were various special arrangements within ordinary firms. Very small firms, with their lesser degree of bureaucratic inertia, were naturally more receptive to unconventional arrangements.

For example, Integrated Software Systems Corp. was a small software firm in San Diego with five employees and one program. The company had been founded about 1970, with the purpose of commercializing[1] some software which one of the principals had developed incidental to his Ph. D. dissertation in engineering at the University of California, San Diego. All five employees worked at home, using computer terminals, and accessing remote computers by night. There was of course no central office-- a waste of money in so small a firm. However, by 1975, the team's chosen meeting place was the home of the one female employee and working mother. Sunni Harris, although she was one of the two newest members of the team, lived in a conveniently central location, relative to the other team members, so they naturally all met at her house. She had previously worked for Lockheed, one of Integrated Software Systems' customers.[2] In other words, from

---

[1] Commercializing software would mean taking some theoretically interesting but hopelessly defective software, and improving it to the point where it was no longer more trouble than it was worth.

[2] E. M. (Edith Myers), "Working at Home and Liking It," <u>DATAMATION</u>, January, 1975, p.103-106

Lockheed’s point of view, working with small innovative startup firms meant exposing Lockheed employees to the work norms of such companies, and tolerating a steady erosion of defecting personnel. This erosion would have to be reflected in Lockheed’s conditions of employment.

The conditions of employment offered by big companies did in fact make considerable allowance for the working mother. Mary Lynn McCaffery, who in 1980 had reached managerial rank at Citibank, had spent fully six years working part-time, circa 1970, while her children were young. At first, she had worked with a terminal, and went into the office once a month; then later, she had gone to a three-days-a-week arrangement. When she was ready to resume full time work, it was all of three to four months before the bank started promoting her into managerial positions.[1]

Thus, when the woman programmer’s children started school, she was apt to find herself still in the programming game. At this point she might have several years of programming experience, and thus be ripe for promotion to manager. For women to make their way into management required them to convince their superiors that they were not going to quit suddenly in order to raise families. When a woman programmer expressed a desire to be considered for a vacant management position, there would follow an earnest discussion, of the variety which Tracy Kidder calls

---

1. "Women in Management," loc. cit., pp.132, 134

"signing up."[1] The higher manager’s attitude in such a discussion could be misconstrued as sexism, but it was probably more of devil’s advocacy. The purpose of the discussion, as it would have been with a male subordinate, was to make sure that the candidate understood all the implications of the new job, in terms of responsibility, obligatory overtime, etc., and was prepared to accept them.

For actual or prospective mothers, discussion of child-care arrangements was necessarily included. Such discussion did not imply rejection of women pro women, or mothers pro mothers, but was merely an insistence that they think through the implications of combining a demanding professional career with motherhood. Christine Millen got her decisive promotion by declaring: "You don’t have to worry about my home life. I’ll worry about that--and I’m prepared to go abroad. I’m prepared to make the extra energetic effort on behalf of the company."[2]

Obviously, such a woman might well need (and be able to pay for) a much more expensive and flexible form of child-care than that appropriate to most working women. At a certain point of advancement, she might even have to get a live-in nanny. But that too would be affordable by the time it became necessary.

Here again, the rate of expansion entered in. Many of the available programmers were simply too inexperienced to be promotable. In the short term, the limiting factor for job

---

[1] The Soul of a New Machine, Avon Books, New York, 1982, pp. 63-66 (orig. pub, Little, Brown and Company, 1981)

[2] "Women in Management," loc. cit., p. 136

opportunities was not so much a limited supply of jobs, but rather the limited rate at which employees could grow to fill those jobs. As long as job opportunities were expanding faster than trained ability to perform those jobs, competition between programmers remained at a fairly limited level, more recreation and learning device than Darwinian struggle for existence.

But of course, in the long run, the supply of jobs mattered too. Programming could expand numerically only if it also expanded conceptually, by continually broadening the range of things which could be programmed. If the range of what could be programmed was extended fast enough, that would provide room for the growth of the total number of programmers, the individual growth of each programmer into a master craftsman surrounded by his or her journeymen and apprentices, and even the growth of labor-saving devices such as advanced programming languages. But if this extension of the range of programming were to falter, then programming would slip into economic stagnation. The expansion inherent in the growth of each individual’s skills would soon carry the profession to the limits of its subsistence. Promotion would dry up, and great numbers of overqualified programmers would appear. People would begin to compete for each other’s jobs, and this would soon become group competition, with racial, ethnic, and sexual groups all striving for their quotas of employment. All this baggage of economic decline could be avoided only by continually redefining programming outward.

Women tended to play a considerable role in this redefinition. They brought more than their share of mental breadth to programming. They were, par excellence, the collectors of new complexity from the outside world to replace that which had been subdued.

Women sometimes introduced humanistic concerns into the discourse of programming. For example, Angeline Pantanges, who in 1977 was an Associate Editor of DATAMATION, and international Editor by 1979, often wrote about the social issues of computing. We find her doing such things as systematically polling the leaders of the field to elicit statements that computers were meant to be about more than corporate data processing, warning against the excesses of European nationalism and English Thatcherism, and supporting Soviet dissidents.[1]

As will be shown later, various women wrote about the issue of women in programming. However, whatever the long-range implications of a resident conscience like Angeline Pantanges, the most important literary role of women in the short run was to write about personal computers. Women did not play a leading role

---

1. "Computer Wish List," (DATAMATION, January 1977, pp. 56-58) and "Social Concerns: Reminiscences and Predictions," (same issue, pp. 156-61) are about the real role of the computer. "No Charter for Human Rights" (ibid., September 1977, p. 244-A) and "Shcharansky's Agony" (ibid., October 1977, p. 150-54) deal with the case of the Russian dissident Anatoly Shcharansky. "Is The World Building Data Barriers?" (ibid., December 1977, p. 90-103) is about the threat of European data nationalism. "They Could If They Would" (ibid., June 1979, p. 194-A) is about the privatizing excesses of the Thatcher government in Britain.

in developing the first personal computers[1], but they were the ones to bring personal computers back into the fold of computer programming.

In the mid to late seventies, advocacy of personal computers was hardly a way to win popularity contests. For one thing, personal computers seemed a rejection of the whole tradition of complexity-managing software which had grown up around computer programming. The first microcomputer sold, the Altair, was actually to be loaded with a program by setting toggle switches, one bit at a time. It was a regression to sometime in the 1930's or 1940's. That was a very brief stage of course. Personal computers soon acquired the minimum necessary complement of peripheral devices, such as keyboards, video tubes (often still a television set), and some kind of mass-storage device [2].

Software, however, remained primitive for some time to come. Systems programming had to be done in assembly language. Application programming was done in what languages were available. The first language available for microcomputers was of course Bill Gates' 4K BASIC, a BASIC interpreter designed to run in only four kilobytes of memory, enough to store about a page

---

[1] Though there were exceptions like Lore Harp and Carol Ely of Vector Graphic, who started their own home business (specifically, in the bathroom of Ms. Harp's home) making a memory board for small computers which Ms. Harp's husband, an engineer at Hughes, had designed for them.

See: Adam Osborne, Running Wild: The Next Industrial Revolution, Osborne/McGraw-Hill, Berkeley, California, 1979, pp. 33-34

[2] Though at first, this did not mean a disk drive, but rather a signal box designed to plug the computer into an audio-cassette tape recorder.

and a half of text. Predictably, the question was not so much what feature this language had, as what features it did not have. It had almost none of the features required to control and manage complexity.[1] There were alternatives languages soon enough,[2] but they had shortcomings of their own, for the same reason. A very small computer could not afford a very powerful language.

For the first few years, personal computers were open to technical scorn. As late as 1980, it was still possible to speak of personal computers in quite unflattering terms. Dorothy A. Walsh remarked that "Personal computers are fun. They are joining hi-fi, do-it-yourself kits, and movies as favorite pastimes. They are useful. But they are not computers in the classical sense." She referred to the personal computer programmer as a "self-made 'programmer,'" and a doer of mathematics homework.[3] Richard Hamming, one of the great mathematicians of computer programing, was even more unflattering:

---

1. This remained the case with more advanced versions of BASIC, up to the Microsoft GW-BASIC of the late 1980's. BASIC programs resembled assembly language in their lack of well defined organization, and their devotion to efficiency at the cost of complexity. The last great program of this genre was Andrew Flugelman's PC-TALK III, some hundreds of lines of maddening rat's nest.

2. The most notable example is Charles Moore's FORTH, which was used to write the VALDOCS word processor. FORTH was no royal road, however. As Marc Steigler and Bob Hansen put it in their Programming Languages: Featuring the IBM PC and Compatibles: "*All* users of Forth are experts: to be able to use the language for anything beyond trivial applications, you must understand both the language and the computer in microscopic detail." (Baen Books, 1984, p.330)

3. Dorothy A. Walsh, "Forum: A Modern Alladin's Lamp," DATAMATION, March 1980, pp. 272-74

> Word processing is getting off the ground; Computers are getting into the hands of secretaries. Authors, and especially frustrated authors, will now produce more and more books with less and less that is new or worth the storage space. God help us.[1]

A somewhat more enlightened commentator, Fred Gruenberger taught computer programming at California State University, Northridge. Even while recognizing that personal computer production was up to eight million machines per year, each comparable to a giant computer of the 1950's, Gruenberger could dismiss personal computer programming as the crude work of twelve-year-olds.[2]

The gibes of people like Walsh, Hamming, and Gruenberger were annoyed responses to the promoters of personal computers. DATAMATION's pages had been infiltrated, not by the core of

---

1.quoted in Gruenberger, below, p. 184

2. Fred Gruenberger, "Reader's Forum: The Incredible Shrinking Decade," DATAMATION, January 1980, pp. 181-84
However, Gruenberger could be extremely sarcastic without actually being hostile in a programmatic way. In an article a couple of years previously ("So You're Trying to Teach Computing," DATAMATION, April 1977, pp. 119-24), he had remarked:

> "Keep in mind that for a student who *majors* in computer science, we might have access to his attention for perhaps 1000 class hours in computing courses. And in that 1000 hours, we will probably have to reteach him how to read, write, and do simple arithmetic. (p. 119)"

Further on, in a quotation from Barry Gordon of IBM, there is a remark to the effect that the student "is rewarded with a good grade (or dog yummy, or small fish, or whatever you throw students these days)" (p. 120). Gruenberger did not have to stay at Northridge-- he could certainly have gone into industry at a hefty pay increase-- so it was almost certainly a case of his bark being considerably worse than his bite.

personal computer builders and programmers, but by people who were engaged in businesses involving personal computers. Far more of them were women than was the case in personal computer manufacture proper.

Word processors were a kind of incipient computer. The better sort of free-standing word processor might have a video terminal and two floppy disk drives, one of which held the word processing program. In short, such a machine was for all purposes and intents a personal computer. Amy Wohl was not the only person to write about word processing at the time, but she was somewhat unconventional in choosing to cover free-standing word processors. While other commentators stressed eventual integration of word processing into a central or networked system of some kind, Wohl promoted the free-standing type, a here-and-now de-facto personal computer.[1]

Still more forceful in her argument was Portia Issacson, co-owner of a computer store, who had a regular column, "Personal Computing," in DATAMATION, starting in October 1977. Her first column began, in a style reminiscent of Abbe Sieyes "What is the Third Estate?" of 1789, with a touch of Karl Marx’s Communist Manifesto thrown in:

---

1. See her "What’s Happening in Word Processing" (DATAMATION, April 1977, pp. 65-74) and her later "Word Processing 1979: A Market Evaluation" (DATAMATION, May 25, 1979, pp. 112-14).

By way of contrast, see J. Christopher Burns, "The Evolution of Office Information Systems," (DATAMATION, April 1977, pp. 60-64) and Robert B. White, A Prototype for the Automated Office," (same issue, pp. 83-90), both of which exhibit the big-system mentality.

> Personal computing is computing so low in cost that it can be used abundantly, even wasted! Personal computing is a revolution in the availability of computing resources. No longer will information processing be available only to large corporations and government; we’ll soon have computing power at out fingertips in out homes and offices. Computers will be dedicated to even simple tasks. The coming widespread availability of the personal computer ranks with other great technology-based revolutions-- the printing press, the assembly line, and the automobile.[1]

Further down, she asked: "What is a personal computer?" and the answer was effectively a restatement of Sieyes’ "The third estate is a complete nation." In this case, personal computing was presumably a complete industry and technology.

Over time, Issacson became more outspoken. By 1979, she was talking about personal computer owners starting to program their computers, and thus being able to launch an autonomous body of software of their own;[2] and calling for the equivalent of a mainframe computer on every desk.[3]

Thus Amy Wohl and Portia Issacson were leading figures in positioning DATAMATION to go into personal computers. By so doing, they helped to prevent the mainline programming community from defining itself into a corner as successive groups of engineers had done over the years. Personal computers would have happened whether DATAMATION and its readers liked them or not, but if data processing centers had consistently stuck to a

---

1. DATAMATION, October 1977, p. 210

2. "1979-- The Year of the Home Computer," DATAMATION, January 1979, pp. 217-18

3. "Personal Computing: 1984’s Information Appliances," DATAMATION, February, 1979, pp. 215-218

hostile attitude and refused to cooperate with personal computer users, there could have been considerably more disruption than there was.

Thus, women had promoted the idea of social responsibility, and they had promoted the dangerously unpopular idea of the personal computer. From there, it was no great leap to promoting themselves, that is, to promoting the idea that women as a group could be programmers, and do well at programming. The public image of the woman programmer gradually caught up with the facts, often running ten years or more behind reality. During this period, women programmers were making immense gains, not in percentages (which had already been achieved), but in absolute numbers. This advance was not noticeably hampered by the fact that the public sphere of the computer press, and especially the advertisements, still treated women in a nonserious fashion.

In the early sixties, sexism was still blatant and outspoken, though even then it tended to attach itself to women who were not programmers. In 1959, a group of RAND Corporation employees, including E. A. Feigenbaum and F. J. Gruenberger, had written, in a letter to the chairman of the Western Joint Computer Conference: "It is ridiculous to have to sweat out a long line, then get your card to fill out, then find that the little girl has never before met the problem of making change..."[1]

A letter of announcement from the COBOL committee, published

---

[1] Feigenbaum et. al. to Robert M. Bennett., 15 July 1959, published in: "And Still More on Computer Conferences: Letters to and from WLCC’s Chairman" DATAMATION, January/February, 1960, pp. 43-45, see p. 45

in the July/August 1960 issue of DATAMATION, made no reference to Grace Hopper.[1] It was written in the same bureaucratic-impersonal mode which had caused Feigenbaum and Gruenberger to use only their initials, and was full of references to anonymous subsidiary committees-- committees which, as we have seen, were about a quarter female. The only name attached to the letter was that of the chairman, even though it may well have been a joint production.

A cartoon of "The Well Dressed Programmer"[2] published in 1965 showed a recognizable grid (ie. gunsight-style) etched on one of the (male) programmer’s eyeglass lenses. The caption explained that this was for "sizing up secretaries."

Women programmers were collectively invisible. The programming of the early sixties did not overflow with public roles, and no doubt each woman programmer, within her own sphere, tended to be

---

1. Charles A. Philips, letter, June 30, 1960, published as: "CODASYL Emphasis Shifting to Development Committee," DATAMATION, July/August, 1960, p.70-71

2. "The Well Dressed Programmer," DATAMATION, Jan 1965, p. 58

treated as a special case-- an honorary male as Antonia Fraser[1] would put it-- with no great awareness that there were enormous numbers of special cases out there. The shortage of programmers ensured them a fair chance to show their ability, and once shown, they could live on their earned individual reputations. However, the material basis for even this kind of ineffectual sexism was being steadily undermined.

A transition point was reached in an unusual little piece about a small plastic model computer, the DIGI-COMP I.[2] The piece was written by a fourteen or fifteen year-old schoolgirl, Karen Ann Schneider. Miss Schneider was by no means the first female DATAMATION writer. Established professionals had gained the recognition of publication in the main trade journal. But she was in all probability the first female DATAMATION writer who could not possibly fit into the category of honorary male. She alternated between the conventional role of stage moppet,

---

[1] In her <u>The Warrior Queens</u> (*), Fraser deals with a type of woman occupying an emancipated role without desiring to emancipate other women. For example, of Elizabeth I (and by extension, Margaret Thatcher), Fraser remarks: "The fact was that the 'dread Virago' herself had never made any effort to improve the general appraisal of woman's worth; for cogent reasons, that was the very last of her intentions... It was customary for her to deride her own sex along stereotyped lines, out of policy."

Thus, an "honorary male" programmer would view herself as quite distinct from the secretaries, and would avoid doing things which might cause her to be associated with them (eg. admitting to being able to type or keypunch).

(*) Vintage Books, New York, 1990; Alfred A. Knopf, 1989; orig. pub. as <u>Boadicea's Chariot</u>, George Weidenfeld & Nicholson, Ltd., London, 1988

[2] Karen Ann Schneider, "DIGI-COMP User Report," <u>DATAMATION</u>, January 1965, p. 55

familiar to watchers of old situation comedies; and something new, a person engrossed with a computer, an incipient hacker. The tone of the short blurb provided by the editors tends towards the tone customarily used to describe children playing at being adults (eg. the formal resolutions of the junior high school student council on the subject of international politics, or something equally beyond their ken). The woman-as-little-girl image was more or less spontaneously disintegrating before one’s eyes as soon as a computer was attached to it. Women programmers were coming to recognize themselves, not as extraordinary women who had nothing in common with ordinary women, but rather as representative women, blazing a trail which other women could follow if they wished.

In due course, women programmers began to develop a modest literature of programming feminism. In the March 1, 1971 issue of <u>DATAMATION</u>, there was a letter from one Pamela A. Henline calling for the recruitment of women programmers, the creation of part time positions for mothers, etc.[1] None of this was particularly new, or as far as one can discover, even uncommon, as far as actual practice went. It was just that women programmers were still thinking of themselves as exceptions. This myth of individual exceptionalism naturally did not survive public discussion. As people pooled their knowledge, they came to realize that there were a lot of women programmers.

There was a long distance between encouragement of women

---

1. p. 13

programmers and concurrence with the full rhetoric of the women’s movement. When Anne Petrokubi reproached the editors[1] for describing a successful woman executive as comely, and for supposedly denigrating feminist militancy, the editors replied, coolly, that the offending item was both written and edited by women. Likewise, Bernard Galler, who taught Computer Science at the University of Michigan, reported on the explosive anger of his female students, when a recruiter gave the impression that they counted as affirmative action hires.[2] This was a far cry from the "right to affirmative action to make up for past discrimination" doctrine put forward by some feminists.

It was Gonnie Siegel who formulated a new public consensus on sexism in programming.[3] Ms. Siegel was a management consultant, specializing in feminist issues, and past president of her local National Organization of Women chapter. It is not clear whether she was a programmer, but she was probably not a very highly qualified one. Her analysis was not really formulated around programmers, or engineers, for that matter. It was more based around the general mentality of the businessman. She attacked discrimination against women not as wrong, but as illogical. Discriminatory managers, who prided themselves on their rationality, were in fact in the grips of another cluster of emotions, which can be approximately glossed as machismo. One

---

[1] Letters, <u>DATAMATION</u>, February 1976, p. 8

[2] Letters, <u>DATAMATION</u>, June, 1977, p. 223

[3] Gonnie Siegel, "The Forum: The Best Man for the Job May Be a Woman," <u>DATAMATION</u>, June, 1976, pp. 196-200

element of this machismo merged into pride in rationality, especially as expressed in ruthlessness. Only by overcoming their sexism and machismo could managers become truly rational.

Much of this was not directly relevant to technical management, and especially not to computer programming, where as we have seen, there simply was not the demographic basis for any meaningful sort of discrimination or abuse of power. Nor is the cult of ruthlessness ever, even in the short run, very workable in technical management. In technical management, the problem inevitably centered on a machine. Machines are impervious to fear, and therefore cannot be coerced. Thus, when a manager intimidates subordinates, the result is more likely to be falsified work, rather than a breakthrough. It would have seemed that Siegel's discussion of sexism in general management was not very relevant to computer programming.

However, what was productive was Siegel's opposition of rationality to the cult of power, as expressed in business machismo and the idealization of ruthlessness. A woman programmer who adopted Siegel's approach would find herself in alliance, as fellow programmers, with her male colleagues. Thus, one of the letters in reply to Siegel's article, from Peter Martin,[1], was about the liberating effects of not having to pretend to "business macho," along with the "ulcers, heart attacks, and other dysfunctions" it would produce.

Feminism was being transformed in the direction of anti-

---

[1] Letters, DATAMATION, September, 1976, p. 7

authoritarianism. Elements of the ideology which worked around computers were being retained, and even generalized to include men, and those elements which did not work were being discarded. Thus, programming feminism as a movement did not have a very extended life.

Even the seemingly promising agenda of pay discrimination was elusive. As Christine Millen remarked in 1980:

> Maybe I was paid a thousand or so less than a man at times, but there was never any substantial differential as far as I could determine... No large corporation can afford to discriminate... [and] in large corporations, you *do* find [them] out.[1]

To this, Molly Nemhouser replied:

> Contrarily, I've worked for smaller companies; information like that has been more secret, more protected. When I found out I wasn't making an equal salary, I asked for more money. And I got it, no problems, no questions asked.[2]

Women programmers might exemplify the highest goals of feminism, but that did not necessarily make them receptive to feminism. To

---

[1] "Women in Management," loc. cit., p. 139

[2] ibid.

the extent that feminism was geared to the aspirations of women who were still in an economically dependent condition, women programmers would find it unconnected to their own lives, and would espouse it only in a perfunctory way, out of a sense of obligation. The main effect of feminism in programming was to mentally advance women programmers to the psychology of post-liberation. They were coming to recognize themselves as they had in fact been since about the year 1960, neither honorary males, nor oppressed victims.

Editorial content appeals to the rational intellect. It was therefore the first to respond to a rational argument. Advertisements lagged much further behind than editorial content. Advertising, even technological advertising makes a greater appeal to the unconscious mind, as opposed to the rational and conscious intellect. Advertising is a sort of collective unconscious. And the unconscious is supremely conservative, capable of running anything up to fifty years behind actualities.

At first, the advertisements treated women in a simple-minded way. For example, there were the Computape advertisements, circa 1962, in which a girl named Penelope flirts with a reel of tape.[1] Blatantly sexist advertisements persisted up to about 1970. Deborah Sojka of DATAMATION later collected an assortment of these advertisements.[2] For the late sixties, there are: a picture of a pregnant woman, with the caption that her mechanical

---

1. DATAMATION, February 1962, p 62; August 1962, p. 74

2. "The Old Clothes of Advertising", DATAMATION, September 1982, p. 137-53

replacement won’t get pregnant and be incapacitated by morning sickness; a United Airlines ad with a pun on the ’stewardess’ or ’coffee, tea, or me’ joke; and the inevitable girl in a bikini. The worst offenders refused reprint permission, obviously hoping that would be the end of it. But Ms. Sojka was not going to let them get away that easily. Instead, she printed their citations and brief (fair use) quotations of their copy. For example one firm had said, in 1970, "We taught our data entry system to speak a new language: Dumb Blonde."

But that was one of the last instances. In the early seventies, advertisements began to catch up with the editorial content of the mid-sixties, and the social actualities of the mid-fifties. Of course the Association for Computing Machinery was one of the first organizations to change over. By 1971, it had used, for a membership recruitment advertisement, a profile of a recent new member, a woman systems programmer.[1] But that was something of an exception, remote from the normal pattern of Id-driven selling.

The more characteristic advertisement of the early-to-mid-seventies was an analog of the editorial treatment of Karen Ann Schneider. That is, it depicted woman in the process of being transformed from little girl to adult by the action of computers.

A Computer Terminal Corporation advertisement suggests: "scare hell out of your secretary. Get her a computer. About 45 minutes after the handsome thing is on her desk, she’ll

---

1. ACM advertisement, DATAMATION, April 15, 1971, p. 86

be an expert." There are a series of time-sequence photographs of the secretary going through a phase of terror, having hysterics, and so forth; but eventually winding up with a delighted smile on her face.[1]

A Digital Equipment Corporation advertisement offers: "...A big computer system that's small enough for anyone," and shows a little girl taking a computer out of the boxes, putting it together, and eventually sitting down in front of it.[2]

Then, having established that a woman could use a computer, the advertisements began to examine the relationship between computers and power. A Prime Computer advertisement[3] depicted four women doing various interesting things using computers. One of them was a hard-core ('real time') systems programmer; another was doing an actuarial simulation; a third was doing a literature search; and a fourth was writing a manual. Interestingly, while the last two are wearing conventional feminine clothing and posed in conventionally demure attitudes (especially the manual writer[4]), the hard-core programmer is wearing slacks, sleeveless tunic, sensible shoes, no make-up, and is standing in an assertive posture which may or may not be a fencer's stance. The statement was

---

[1] <u>DATAMATION</u>, March 15, 1971, p.43

[2] <u>DATAMATION</u>, July, 1977, pp.38-39

[3] <u>DATAMATION</u>, August 1976, p. 38

[4] A sometime woman programmer, to whom I showed this advertisement, used the expression, "Waiting to be kissed," to describe the manual writer.

quite clear: there was a place for women who wanted to remain conventionally feminine, and quite a good place by most standards, but there were also additional opportunities for those who would discard all the traditional paraphernalia of ladyhood.

As the seventies came to an end, commercial advertisements rapidly trended towards the tone the ACM had adopted at the beginning of the decade: matter-of-fact, with equality taken for granted. A Bell and Howell advertisement showed a woman troubleshooting a printer with the company's new recording oscilloscope.[1] An Informatics Inc. advertisement offers a testimonial from a woman systems analyst.[2]

In one of the most striking advertisements, Intertel, a maker of modems and computer networking devices, turned the very idea of sex appeal on its head to make a nasty joke at the expense of a rival. They showed a picture of the kind of sequin-clad blonde who might have starred in a daytime game show, and commented:

> The 1960's. A face. A smile. And buy, buy, buy... You could take some starstruck kid from Kankakee. Dress her up like a fashion model... And every time she smiled, cash registers all over America would start to chime... It wasn't just toothpaste, either... Take the folks at Milgo, for example. They sold modems that way. A lot of modems.
>
> But all of a sudden, the sixties were gone. And the 1970's rolled in like a wave. People discovered they had networks. Not just modems and terminals (They also discovered that women weren't sex objects, but that's another story)... While Intertel was busy building network control systems, Milgo kept right on building modems... And they waited for the orders to roll in. While they were waiting, Intertel introduced them to

---

[1] DATAMATION, January 1976, p. 135

[2] DATAMATION, January 1979, p. 19

the rough and tumble world of marketing in the 1970’s.[1]

In short, sexism is not an exercise in robbing women of their rights, but merely the mark of an incompetent fool. At that point, sexism in computer advertising was unequivocally dead. It was not a matter of "oppressive speech" being banned. Rather, sexism had simply been shown to be hopelessly at variance with reality. It was not politically incorrect, but factually incorrect. An insistence on the false premise of sexism would, by quite correct inferences, entangle one in a series of other unrealities, as well as the disasters which go with departure from reality.

By the year 1980, when Gloria Steinem was writing her fantasy about cleaning women, women programmers had attained a series of degrees of freedom and equality. They had become numerous. They had become economically independent, and if pay scales for women programmers were not equal yet, equality was fast approaching. Women programmers had found autonomy in their work, either as highly skilled professional employees, or by running their own companies. They had attained both individual and collective respect within the profession, at both the conscious and subconscious levels.

This victory had not been attained by affirmative action or quotas, nor by political agitation, nor by efforts to ban the expression of derogatory sentiments. It had been gained by harnessing the potential of the computer. Each development in the

---

1. DATAMATION, March 1977, pp. 10-11

computer’s technology had implied a social correlary, and on the whole, the correlaries had run in the direction of human freedom.

Now, in 1980, another computer revolution was impending, that which Portia Issacson had proclaimed. The next phase would be that of the personal computer, and it, too, would have its correlaries.