425 Portland Avenue
Minneapolis, Minnesota 55488-0002

StarTribune

MINNEAPOLIS • ST. PAUL

Dear Efrem Lipkin

I finally found that Community Memory piece I promised to send you - sorry for the delay. I hope it brings back happy memories.

Best,

Jeremy Iggers

Final Draft of Tom's Article
OK'ed By Dan Leer...

CANDY

HIGH-TECH ALTERNATIVISM -- The Case of the Community Memory Project

Tom Athanasiou (415) 826-0128

David Noble recently argued, in the now defunct democracy magazine, that "the fight for alternatives...diverts attention from the realities of power and technological development, holds out facile and false promises, and reinforces the cultural fetish for technological transcendence." (1)

I'd like to agree, and yet take issue with Noble's conclusion -- that political activists concerned with science and technology should concentrate almost exclusively on strategies of opposition.

Anti-technological confrontation is crucial, but alone it cannot support the development of a stronger and more sophisticated technology control movement. Confrontation should be judged as much by its success in catalyzing larger cultural and political shifts as by its immediate effectiveness in blocking technologies of passivity and death.

The point is not to stop "technology," but to make and enforce different choices about it (2). The popular political culture that can inform such

----

(1) Present Tense Technology was printed in three parts, in the Spring, Summer and Fall 1983 issues of democracy. It demands reading, for in a period characterized by a rather blithe leftist realpolitik, Noble calls for a focus on the simple truths of social power and class antagonism. His assertion that the "friends of labor" disarm the working class by exaggerating the possibilities for workplace technologic reform must be taken seriously.

Still, this article will take a more agnostic position, reflecting as it does the authors inability to imagine a modern radical technology movement without an alternativist dimension.

(2) In this regard the women's pro-choice movement is perhaps an evocative model of a social movement that, while concerned with technology (in this case medical technology), is oriented more towards new options than to direct opposition. (When is comes to abortion, it's the right that's "anti-tech.") The "choice" issues raised for pregnant women, in significant part by the availability of technological options, highlight cultural and political dilemmas which cannot be addressed by simple opposition. In many cases (i.e. nuclear power) simple opposition is clearly the appropriate response to capitalist technological innovation, but with regard to micro-electronic and biological technologies it will not do.

choices will be built not only from the passion of refusal; it must express as well the widespread fascination with modern technology. Only when that fascination is brought up against the dark side of todays technological constructions will we have in hand the core elements of a new technical culture.

Many technologies have been passed over, ignored and suppressed in the last few hundred years. The alternatives movement of the 60's and 70's concentrated, for good reasons, on demonstrating the viability of alternative approaches within the relatively low-tech worlds of energy production and agriculture. Today this will not do; the environmental sensibilities of the 1970's have been overrun by neo-liberal "tough-mindedness." Opportunistic theories of "re-industrialization" have colonized the political spaces opened up a decade earlier by movements with far more substantive intentions. Today the alternatives movement, if it is to remain politically relevent, must find itself anew in the debates about computerization (3).

The microchip is nothing if not flexible. Applications abound, but capital will pursue only those which show promise of profit, or which provide for more efficient systems of control. It will continue to repress those potentials which run counter to it's overall logic of commodification and pacification.

One of these potentials is the development of a "de-massified" mass-media. Such media would be based upon the integration of computer technology into electronic communications systems in such a way as to support a wide range of active individual

----

(3) High-tech alternativism carries with it the danger of strengthening science-based ideologies of "progress." Such ideologies, eroded over the last few decades by the insanity of heavy-metal capitalism, threaten a resurgence in the guise of the "micro-electronic revolution." The constantly insinuated image of the micro as toy, helpmate and liberator has the effect of rejuvenating technologic wonder as a crucial element in the systems of social apologism.

As is usual whenever the mass imagination is riveted to the gleam of the latest trinkets, capital benefits.

On the other hand, any politics of technology that declines to focus exclusively on opposition must propose alternatives to systems designed and implemented around the imperatives of Capital. Cooptation is no danger as long as nuclear war, or technologically structured passification, or the corporate production of cancer, remains the focus of attention. But attempts to, for example, define alternative models of industrial automation, attempts which are becoming increasingly popular among left economists, (Cooley, Shaiken and Melman, just for starters) must risk absorption into the same sort of humanistic high-tech ideology that confronts Community Memory.

initiatives. They could provide for the elimination of the distinction between producers and consumers of information, encourage public conversations and, by virtue of being embedded within community social institutions, empower rather than pacify their users.

If we ever make it out of the historical cul-de-sac of capitalist society, such media as this will certainly play a role in the organization of public life (4). Still, their present value is a matter of some uncertainty, and is consistently overstated by its proponents. Certainly it is true that computer communications is widely described as "revolutionary" by many who hardly have the end of institutionalized domination in mind. Like solar technology in the 1970s, the notion of democratic computer communications attracts more than its share of the politically tame. Nevertheless, the liberatory potential of solar power will not be realized during the age of Capital, and the same is true of information technology. The impulse to develop non-hierarchical communications systems (not market oriented videotex systems) contains a real moment of rebellion against the hierarchical logic that today dominates technical design.

II) THE COMMUNITY MEMORY PROJECT

> "Community Memory is a system for the public management of public information. It is an open channel for community communications and information exchange, and a way for people with common interests to find each other...All of the information in the Community Memory is put in directly by the people who use the system: anyone can post messages, read any of the other communications that are there, and add comments or suggestions at any time. (5)"

The Community Memory Project in Berkeley is an attempt to build and deploy such a non-hierarchical computer communications system. I have been active in the project for many years, and have developed strong opinions about both it's efficacy and it's history. This article reflects those opinions more than any group consensus, though it has been read by many others, and I fancy that most would agree with the bulk of it.

----

(4) For a relatively recent example, see the "Kenner" network in the leftist utopia Woman on the Edge of Time by Marge Piercy. Fawcett, N.Y. 1976

(5) From An Introduction to Community Memory, available for one dollar from Community Memory, 916 Parker Street, Berkeley CA 94710.

Concieving of such a project is relatively simple; carrying that conception into reality is a different matter. The execution would take a lot of time, energy and money -- large software projects are very expensive to develop even when the results are very cheap. The founders knew this, and tried to take it into account. They decided to proceed by writing the underlying "system software" in as, modular a way as possible -- in effect implementing the Community Memory system on a generalized text/data handling "toolkit" that could support commercial spin-offs as well as Community Memory itself. These spinoffs wouldn't take much time, and they'd provide the reams of cash necessary to support the system.

So the story of Community Memory is really two stories, reflecting our history as a political/technical collective that took' a long unplanned, and largely unpleasant trip through the computer industry.

Only in the last year have we actually completed an advanced prototype of the Community Memory system itself. (The first three public terminals will be operational in Berkeley in July of 1984.) Consequently, our experience can hardly be the basis of overblown generalizations about high-tech alternativism; we've learned only how difficult it is to beat capitalism on its own terms. We've little experience in actually operating a non-hierarchical communications system. We don't know yet whether our efforts were worth the trouble. We may never know -- or agree among ourselves.

People, Places, Things

Community Memory has been jokingly referred to as a "closely-held collective." Simultaneously a political organization and a systems software shop, and run by the active participants in a endless series of open meetings, it nevertheless maintains a legal structure which restricts ultimate power to the active "members." We have voted only twice, yet these votes were significant. They reflected antagonisms rooted in our ill-fated venture into commercial software production.

While we've barely succeeded in finishing the first cut of the Community Memory system, we have managed to become a well known and frequently cited populist computing center. And through close association with the publisher of a respected if small-circulation and infrequently published journal, the Journal of Community Communications (6), we've had some success in broadening the typical critique of information technology to include communications issues.

----

(6) The Journal of Community Communications, available from Village Design, P.O. Box 996, Berkeley, CA 94701, is a small, infrequent quarterly that focuses on the theory and practice of "non-hierarchical communications." Many, but by no means all, of the articles are related to computer

Tuesday night dinner/meetings are usually attended by about a dozen, but there are altogether about twenty "friends of Community Memory." We are writers, still using some of the Left's first word processors. We are programmers who would rather not work for the military -- or the banks. We are ecology and peace activists. We are ex-hippy marginals now earning good wages as technical writers. We have become alternative communications experts; one of us even has a degree that says so. We are, as a group, sharp, scientific, and articulate. We even have our own sociologist, a lucid and funny fellow from the UC Berkeley sociology department that has been "studying" the project for several years now.

All this to do again what had already been done. In 1973 a small group of technologically astute activists in San Francisco put up a small three-terminal Community Memory system and kept it up for about fourteen months. Uses reflected the locations of the terminals. One was in a music store and collected information about gigs, bands and the like. Another, at a hippy hardware store, specialized in Alternative Technology and barter. The third, located in a public library in the Mission District, a poor area of San Francisco, was little more than a high-tech graffiti board.

The experiment was pronounced a success, but attempts to raise the funds to expand and continue it proved a failure. Years later a group including some of the original founders decided that conditions had changed enough to justify another, more ambitious, pilot project. This meant writing the software from scratch, and writing it to run on the new generations of supermicrocomputers that had become available in the interim. The new system would be powerful, sophisticated, and most of all, portable. It could be replicated again and again, in different neighborhoods in different cities. Or it could be used by "nongeographical communities" of people who had specific needs for a dynamic shared information pool, e.g. community organizers from different regions. All this would be possible because the new system would eventually tie individual Community Memory Systems together into one grand "network."

communications. Coffee-houses in 16th century London, a Lakota indian radio station, pornographic software, Control Data Corporations' venture into Appropriate Technology and Punk Rock have all found space in recent issues.

Pacific Software

When we failed to find marketing company already in existence we looked to our own circle for people to handle our commerical spinoffs. In this context the fact that we could count among our friends several successful inventors and small-scale entrepreneurs was seen as an asset. Our friends would found a software company, Pacific Software, to which Community Memory would license its goods in the hope of earning royalties to help fund its public programs. Community Memory itself would be protected from commercial entanglements and we could blithely proceed as left-libertarian information activists. Everyone would be a winner.

For the record, the commercial spinoffs we developed were:

1) Sequitur, a sophisticated "relational" database management system built upon the Community Memory "toolkit." Designed to run on small computers, it is distinguished by the high degree of integration that it achieves between text and data processing.

2) X.Dot, a portable C-language implementation of an international standard data-communications protocol, X.25. This is the foundation upon which communication between independent Community Memory systems will be based.

The disinterested observer will perhaps not be surprised to learn that matters did not our proceed as smoothly as we had imagined, and that our subsequent organizational development reflected the contradictions of our political, technical and entrepreneurial entanglements. To some extent, this was anticipated, but we overestimated our ability to manage the contradictions. Years after we began technical development, we have only barely managed to get a functioning Community Memory System out the door.

For the collective, commercial pressure became a mutagenic, almost fatal force. Nor did our friends do well as capitalists. We, for our part, seriously underestimated the market pressures which would keep us from making a fortune and quickly exiting. Pacific Software, for its part, made the situation worse by proving incapable of proper planning and scheduling. There were always changes to be made to the code, always more features that the market demanded. And providing them remained, for far to long, the responsibility of reluctant Community Memory programmers. Meanwhile, Pacific Software went broke.

In retrospect, our collective was organizationally inadequate to the tasks it set itself. Power relations, and with them lines of responsibility, were too informal to adequately replace the managerial mechanisms by which traditional capitalist firms direct operations. Not that they were entirly informal: the small "junta" of long-term and ideologically committed members could not pass without notice; besides, they paid the bills. But since their authority was neither embodied within formal organizational relations nor considered legitimate, it could never be decisively exercised.

We eventually adopted a committee structure, with a "co-ordinating committee" empowered to manage but not set policy, but by then it was fairly late in the game. Our formative years were spent toiling under "the tyranny of structurelessness," a tyranny that obscured crucial realities within a haze of emotionally charged and even arbitrary authority.

Personnel disputes could consume tremendous blocks of meeting time, and yet remain unresolved. Technical reviews of code were constrained by the impossibility of staff changes. Accountability was vauge, with all decisions subject to perpetual renegotiation. Long term planning was simply impossible.

And day by day, market pressures grew.

South Africa and the Naval Surface Weapons Laboratory

We anticipated that capitalist relations would press against our democratic ideals. Yet our anticipation was vauge, and not really to the point. When reality caught up with us, we were startled by its brazenness.

About three years ago, X.Dot was finally ready to sell. Pacific Software was trying to establish it in the telecommunications market, and we all had great hopes. (At the time we were asking about $300.000.00 per copy.) But when Pacific finally landed a customer it was a company in Johannesberg -- they were building a airline reservation system and needed an X.25.

We refused the sale, then went on to a complex and difficult debate that led to our prohibition of sales to South Africa. (Sales to Eastern Bloc countries were already prohibited by the government.) We also developed a method of restricting sales to the military by limiting Pacific Software's sublicensing rights to "commercial and non-proprietary" applications.

Though none among us wanted to make a sale to South Africa, the discussions were complex and tense; the beginning of the end of our naivete. Only one person, who felt that we needed the sale to establish X.Dot in the market, was willing to argue for it. The argument was raised that Israel was as bad as South Africa (indeed the two countries have jointly developed a variety of nuclear arms), but was defeated in the name of political realism. After all, there is an international technology boycott against South Africa, but not against other unsavory states like Israel. We even considered allowing Pacific Software to make the sale and donating our share to the liberation movement in South Africa. But when contacted, a front group for the African National Congress refused the support, preferring to forego the funds and deny South Africa the technology.

I, along with others, opposed the sale on the grounds of its unique benefit to South Africa. X.Dot, designed to be easily transportable amoung a variety of machines, would have been especially useful to South Africa, suffering as it does a technological boycott of some significance. This same criteria of uniqueness led me to support sales of Sequitur to the military, when not much later they became an issue.

It was the disagreement over these sales that first emerged as open conflict. In the absence of the clearcut ethical/political imperative provided by a pre-existing boycott, unanimity broke down. There was reason to doubt, whatever sales we denied ourselves, that our denial would have an effect. Sequitur was of no unique benefit to anyone; there were dozens of systems that could do the job equally well (perhaps better). Furthermore, there was a strong feeling that, whatever military sales policy we were eventually to adopt, it would be necessary to keep it quiet. The logic of the market, many felt, dictated that we not endanger Pacific Software by making it appear political or constrained in undefined ways.

Our shared understandings were breaking down. Just what was so different between doing business with the military and doing business with the banks, between fast death and slow death? In the absence of a willingness to make a public statement, and considering the onerous uses to which information management systems like Sequitur were put in the commercial world, such distinctions as could be made failed to win easy consensus. Instead, the military debate became the locus of expression for more generalized antagonisms rooted in the political and interpersonal ambiguites of the larger venture.

With the system itself so far from realization, it could only enter our calculations as an abstraction. Anti-military initiatives, in contrast, were very much part of the political environment. The military debate was fertile ground for an organizational crisis for exactly this reason; it forced an end to the easy assumption of political efficacy. The value of the project now had to be weighed against more concrete realities.

Just why was Community Memory a valuable political project anyway? Was it because it would try to demonstrate that high technology could be used to counter social alienation? Was it because by doing something we would be taken seriously even in the pragmatist nightmare of modern America? Was it our institutional existence itself that was important -- that a few lost souls of the computer state would hear of us and know they were not alone? Was it because we were struggling with radical collectivism? Or would the system itself justify our small contribution to capitalist modernization?

It was a long debate and many claims were made. Of them all, one seems, in retrospect, particularly problematic -- the claim that "all money is dirty money." I supported this position myself, and argued that the market would squeeze us dearly for our silent protest. In the end I was over-ruled, and have come to be glad for it. The abstract consistency that I wanted had no place in the contradictory reality we had stumbled into. I remember it now as determined as much by frustration as by political considerations themselves,

Labor-Process, Work Identity, Democracy

Programming labor always held a special status at Community Memory. Despite occasional conscious and semi-conscious attempts to value non-technical work more highly, the logic of the project itself imposed the centrality of programming labor. In the last instance, it was the success or failure of the programmers that would determine the success or failure of the venture as a whole.

As technical development became absurdly protracted, the social consequences of the situation emerged more clearly. We fought the tendency for non-programming staff to become mere support staff for the programmers; such a situation would have been intolerable, especially since only one of the women was a programmer. But we were never able to break the hegemony of technical work over the project as a whole. Worse, the larger group never managed to get control of the design process. It was "technical decisions" made by programmers, decisions about arcana like program capabilities and software development strategies, that led us all into the labyrinth.

And the programmers group was not without its' problems; we had our own meetings, our own work-process, and our own struggle with the old guard. During the early days we established a design process based in an odd mix of extreme heirarchy and participatory democracy. One of the founders, Mr X., played the role of "chief programmer" in our democratic appropriation of the traditional "chief programmer team" (7). The rest of us, programmers of various skill levels, played indians. Mr. X. came up with basic designs; we discussed, criticized, redesigned, and coded them.

To a large degree it worked: we were able to appropriate the broadly varied skills and predilections of individual programmers within a design/implementation process that both preserved the thematic unity of design (Mr. X. did most of it) and allowed for genuine participation and learning on the part of those not playing the role of "chief programmer."

----

(7) In a traditional "chief programming team" the work process is organized to maximize the productivity of one super-programmer, by factoring off as much of his (sic) work as possible. The chief-programmer will do all the design, and most of the programming, but he will have "coders" to which he can assign fully specified sub-modules. The team will also contain a librarian, a secretary, a hardware support person, a backup chief programmer, etc.

But as time went on and pressure mounted, the tenuousness of our accomadation became more manifest. Mr. X. lost his commitment to pedagogy and his willingness to have others challenge his designs. Further, he became increasingly unwilling to stop designing, and to work towards a final artifact. Others, myself included, found ourselves increasingly in the role of technical managers: trying to impose schedules and accountability, maintain a functional programming environment, and continue to integrate and debug the system as a whole. Others refused responsibility altogether.

I mention all this to underscore the fragility of our democracy. We were able to to subsume skill differences within co-operative relations, but the understandings within which we did so were not robust enough to survive the long campaign. Here too, as in the larger group, the insufficiency of our work democracy manifested itself in the breakdown in personal relationships. Some of us were, of course, more trouble than others, but these contingencies of personality were the means by which the inadequacies of our process manifested itself, not the causes themselves.

In the end, Mr. X. and two of his closest compatriots left the group. Those of us who remained were largely in agreement about what had happened and why, so it's no surprise that we were able to continue to work together. Indeed, with the departure of our most design oriented member, we were finally ab'e to concentrate on the more prosaic, but now absolutely crucial task of finishing the programs.

*****

It's taken me some time to achieve a reasonable perspective on these events. I remain unsure of the relationship between our programmer team/work collective and more the typical forms of technical work organization. Many assure me that there is no difference at all, and that many, if not most, programming projects are run as loosely as was ours. I remain unsure.

Certainly our articulated commitment to participatory democracy made for significant ideological differences, and certainly too our programmer/manager relations were somewhat atypical. But are there deeper similarities? Non-Taylorist forms of organization have often been found necessary in skilled technical work, so we can claim no uniqueness here. And what of the "family feeling" so many Silicon Valley firms strive to cultivate amoung their staffs? Wasn't our "collective" analogous here as well?

A really severe interpretation of Community Memory's history would reveal even more disturbing parallels with typical high-tech paternalism. Instead of management we had "the Junta." Like management they kept organizational goals firmly in mind, like dad they defined the norms by which the children were judged.

And like typical professional workers, we worked as much for our identities as for our articulated goals. Like a position in the typical corporate family, a job at Community Memory came complete with a sense of place. Only in this case that sense was defined not in the terms of fast-lane achievement culture, but against them. We were radical engineers, community designers, peoples programmers.

III) WHO CARES ABOUT DEMOCRATIC COMMUNICATIONS ANYWAYS

Today, five years after the reincarnation of Community Memory, it's obvious that our fundraising strategy didn't work out quite as planned. But it did make sense at the time, and our lack of luck with it doesn't mean other groups elsewhere couldn't do better. But one warning is certainly appropriate: we thought ourselves very sophisticated, but we failed to be sophisticated enough to manage the pressures of the market. Given enough time, we will see forms of work democracy far in advance of any existing today. But will they exist and prosper under the regime of Capital? Our experience indicates that they will not.

We have accomplished something of our original goals. We've done a lot of writing, and gotten a lot of publicity. Perhaps we've even had some small effect on the norms by which computer communications systems will be judged. The first version of the system is working fine in our shop, and by the time this article sees publication the terminals will be out; one in a community center, one in a general store, one in the Berkeley Co-op. Beyond that our future is unwritten.

Conclusion

Community Memory's stated and most obvious goal is to demonstrate that computer information systems can be built that will help people to meet other people with similar interests, in effect to create an electronic public space. This modest intent is not anti-capitalist, except in that manner that all "useful utopias" are, by invoking concrete images of alternative futures, thus making the end of this miserable world more easily imaginable.

The intention of this essay has been to argue for the importance of an alternativist aspect within the larger technology movement, and to trace the history of one benighted attempt to make that argument in practice. It should not be read as an overstated plea for an "alternativist" politics, for such a politics will certainly fail if left on its own.

Within the present ideological climate, it's very difficult to project a concept of technology that is both visionary and critical. Community Memory is important because it tries to do just that. It is visionary because it demonstrates, in a concrete way, that we can design new technologies to serve our own chosen ends. It is critical because it contrasts itself with the productions of the telecommunications corporations, and challenges their reduction of human social interaction to the passive consumption of information commodities.

This is an odd moment in human history. It is difficult to imagine a revolution without hope, and today hope requires the healing of not only nature, but technics as well. Somehow the refusal of capitalist technology must simultaneously affirm the possibility of shaping tools to other purposes.