ISSN 0105-8517

The Head and the Heart

Joan Greenbaum

DAIMI PB – 237
December 1987

AARHUS UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
Ny Munkegade 116 – DK 8000 Aarhus C – DENMARK
Telephone: + 45 6 12 71 88 *Telex: 64767 aausci dk*

The Head and the Heart

Using gender analysis to study the social construction of computer systems

Joan Greenbaum
Computer Science Department
Aarhus University
Denmark

Abstract

In recent years there has been a good deal of healthy experimentation with system development methods and work organization, particularly within Scandinavia. This paper attempts to go further into the questions of the development and use of computer systems by using a gender analysis of the issues. Specifically, it examines the organization of labor and patterns of communication used in developing computer systems. It suggests that the use of gender-biased dichotomies strongly influences both the questions system developers ask and the way questions are asked.

Note (1)

This paper was prepared following extensive discussions with Randi Markussen and Gitte Møldrup, Department for Information and Media Science, Aarhus University, Denmark. It was originally written for the 8th EGOS Conference (European Group for Organizational Studies), July 1987. The author is on leave from LaGuardia College, City University of New York.

The Head and the Heart

"The most immediate issue for a feminist perspective on the natural sciences is the deeply rooted popular mythology that casts objectivity, reason and mind as male and subjectivity, feeling and nature as female. In this ***division of emotional and intellectual labor,*** *women have been the guarantors and protectors of the personal, the emotional , the particular, whereas science--the province par excellence of the impersonal, the rational and the general--has been the perserve of men."*

Eveyln Fox Keller (2)

For ten years I worked as a programmer and systems developer. While I enjoyed the work, I was often bound up in the problems and conflicts associated with being a woman in a 'man's world'. Then, for the next ten years I studied the work of systems development, using new glasses to reflect back on the division of labor in the art and science of developing computer systems.

But the division of labor I had experienced and the division of labor that I studied were indeed on different dimensions or planes. Using the tradition of Marx and Braverman, I examined the organization of labor as the division between the 'head and the hand'--or the separation of tasks of conception from those of execution. (3) What I had experienced, was yet, another dimension of division--that between the *head and the heart*. The search to better understand this division has been a difficult and emotional one. The story that unfolds weaves together a cloth made out of pieces of gender analysis applied to the fabric of computer system development. The cloth may still have some holes in it, but it is better to hold it to the light than wait until a whole new cloth can be woven.

In the tradition of Evelyn Fox Keller, this separation of "intellectual and emotional labor" leads us to ask new questions about methods of work and the organization of labor. Questions that don't just have to do with divisions between work done by men and women, but rather, questions that permeate the way computer systems are developed and the end result of that process- the use of the computer systems themselves.

The system development process is a labor process that results in far-reaching changes in the way people work and the organizations they work in. As technology is introduced in an organization, it is molded and shaped by the "technical rationality" built into the systems development process. Systems development is both a process and a product. When we look at it as a *process*, we examine the methods, the organization of labor and patterns of communication used to design and create a new system. When we look at the *product*, we are seeing not just a new system, but the *way the system is used* and the way people and organizations react and change in the environment of the new system. This paper and the story it tells will focus on the *process* of systems development.

In the last few years computer scientists have begun to examine both the products and processes of system development with an eye toward better understanding the ways that systems interact with people and organizations(4). In doing so they have begun to unravel some of the social, political and ideological perspectives that drive the development process. But to better understand computer system development and use, it is also necessary to zero in on some of the deep-seated assumptions that underlie the development process. For it is these assumptions, unspoken and often unconsciously exercised, that guide the process of development and forcefully influence the way technology is used.

Trying to apply a gender perspective to examine these assumptions is not an easy task. As Evelyn Fox Keller points out in her pioneering work on Reflections on Gender and Science, centuries-old mythology "imposes a veil over these practices".(5) To begin to lift this veil we need to take a closer look at some of the gender myths that have shaped the natural sciences and the technology that springs from them.

It is not my purpose to use a gender analysis to ask questions about the role of women in the system development process. Rather, my intent is to show how the system development process builds on the base of the natural sciences, and in so doing, borrows from the gender-based myths inherent in the sciences. System developers are both educated in the scientific traditions of "technical rationality", and, of course, socialized in the gender-bias that shapes our world. This double process makes it all the more difficult to lift the veil, for the process of developing computer systems remains steeped in the language of technical rationality. A language that (unfortunately for the problem at hand), I have learned to use too well. So, if my language faulters, please bare with me, for the veil that separates the "head from the heart" is one that we have all been told to wear.

Dichotomies and Myths

Fox Keller points out that the traditions of science have historically been rooted in a language that places value on words like objectivity, reason and impersonal judgement. These terms are associated with the way "good science" is done, and through the socialization process, they are strongly connected with the identification of masculinity. In the pressure-cooker of growing up, men are expected to act in these well-defined ways. The other side of this language puts subjectivity, emotion and personal feelings into the world called feminine. If a "real man" acts impersonally, forcing his emotions back into his body, a "feminine" woman moves with her emotions flowing forth.

Of course, on an intellectual level, we know these dichotomies to be patently ridiculous. Men are no more easily pushed to the "objective" pole than women are to the "subjective" one. But it is not our intellectual selves that dictate this dilemna. For it is precisely this problem that Fox Keller calls the "deeply rooted popular mythology". Myths operate within a culture at the level of often unspoken assumptions--feelings and images that remain burried beneath the language of intellectual scrutiny. Yet it is the existence of these dichotomies that influences our thoughts and actions. Let's take a closer look.

Using the language of gender analysis we find that usually a different set of values is identified with masculinity (science) and femininity (nature). For example:

MALE (science)	FEMALE (nature)
objective	subjective
reason	feeling
impersonal	personal
rational	emotional
power	love
things	people

Clearly, the list could go on, for we know, only too well, the sides of the dichotomy we are pushed toward as we wend our way toward becoming adults. What is critical to the analysis here, is not whether or not we *choose* to think or act in these ways, but the values and judgement placed on them. "Good system design", like "good science" falls on the male side of the dichotomy. While we know that "science is a deeply personal as well as social activity"(6), the *process and methods* of creating science and technology strongly reflect those values held to be "male".

These values influence the things we choose to study as well as the way we study them. In her study of scientific activity, Fox Keller tells us:

> "*Judgements about which phenomena are worth studying, which kinds of data are significant--as well as which descriptions (or theories) of those phenomena are most adequate, satisfying, useful, and even reliable--depend critically on social, linguistic, and scientific practices of those making the judgements in questions.*" (7)

But there is more. For not only do these values lead us to the kinds of questions that we ask; they also affect the way we ask the questions. The very existence of dichotomies propells us toward the binary or polar nature of the way computer system development is done. The *process* of systems development is imbedded in methods that force binary thinking. We turn now to look at some examples of methods and work organization in systems development.

Systems Development Methods

Systems development is taught as a process of breaking larger problems down into manageable and controllable descriptions(8). Systems analysts are trained to *identify* problems and *describe* objectives and tasks that need to be done. Traditionally, the systems approach includes methods that help identify and describe objective criteria. Commonly used tools like system flowcharts and system specifications further reduce the choices to a series of step-by-step actions. Within the traditional systems approach, the system developer, him or herself, is seen as someone outside of the process--someone whose impersonal judgements can evaluate the actions of computer users.

The whole process appears logical and concrete. The language of computer system development dictates that computer systems are *complex* systems, and as such, require clear-cut methods to separate parts of the complex problem. As a system developer, I know the usefulness of this kind of reasoning and find comfort in concrete methods that make order out of chaos. Yet, I also know that all too many systems fail to take notice of events that happen on the other side of the gender dichotomy--events that can't be easily squeezed into objective identifications or descriptions. The simple fact that the system developer is seen as someone outside of the problem, puts the emphasis on describing *things* rather than on the *relationship between people.*

The systems approach generally defines *information* as identifiable and *quantifiable* data. Systems analysts are trained to define data and follow the flow of documents--the "paper trail" of office procedures. In doing this, the methods focus on procedures and data; concrete "things" that lend themselves to clear-cut descriptions. In an account record-keeping system, for example, data categories like 'customer code' and 'amount-owed' are the kinds of specific data fields that systems developers are trained to look for.

Procedures and data analysis are useful for describing tasks, but they leave out the parts of the picture that bring communication and human interaction into focus. How do people communicate with each other? What are the informal relationships among the people in the workplace? What languages do they use to communicate? How does their interaction change the nature of the information they use? (9) Turning the spotlight on these type of questions develops another picture of information use in organizations. For information is more than quantifiable data when it gets shaped through human communication.

Critics of the traditional systems approach, particularly within the Scandinavian tradition(10), have begun to ask new questions and look for new methods to focus on missing parts in the development puzzle. Yet, the problem remains, that even the search for new questions and methods will be a difficult one, when grounded in the dichotomous structure of "technical rationality". It is not by accident or oversight that systems development focuses on objective, impersonal, quantifiable methods. The tradition of using these approaches lies in centuries of practice that label these methods as "good science". If we choose to overlook the fact that "good science" is closely bound up within the cloth of gender myths that identify masculinity with objectivity, then we will continue to be suprised by our "mistakes".

Organization of Systems Development Work

During the 1960s and '70s the organization of the computer labor process was marked by increasing division of labor. System developers worked within data processing departments and ventured forth to meet with computer users, who were identified as separate groups in need of the professional knowledge of the computer specialists. Even within data processing departments, systems analysts were separated by function from programmmers, and different types of programming functions were further segmented by job title(11). In part, this division of labor was rooted in the fact that the systems development process was seen as the need to control large-scale projects for mainframe-based computer systems. Indeed, most of the work was done within large organizations that were essentially bureaucratic and organized within formal hierarchies.

Much of this type of formal division of labor has begun to disappear in recent years. The use of microcomputers with pre-packaged software and the introduction of fourth generation languages has helped patch the divided labor process back together. In addition, it began to be apparent that the rigid division between systems developers and users, and systems analysts and programmers, simply did not work. As computer users gained some knowledge, managers found that they could no longer control the divided work process. And as programming became less time consuming, it was evident that the emphasis had shifted to the problems of system design. Now, teams of programmer/analysts are more common, and many organizations are experimenting with integrating system development teams within computer user organizations.(12)

While the recent integration of labor seems much more effective to the development of user-oriented computer systems, it still remains trapped in dichotomies that strongly affect the way work is done. Dichotomies that reflect the historical patterns of gender-bias in the methods and languages used.

Returning for a moment to the discussion of system development methods, it is obvious that the choice of methods directly effects the way work is done. System methods generally rely on step-by-step procedures that result in written descriptions and system specifications. These documents, in turn, represent objective ways of evaluating the product of the system development process. The existence of formal descriptions and specifications is not, in itself, a problem, for system developers could no sooner throw away such documents, than could lawyers, in practicing their profession. Our social, economic and political system is, after all, rule-based and not grounded in tradition. Yet the process of developing specifications, and indeed, their very content, effects the way work is conceived and organized. The documents serve as milestones that mark the territorial boundaries between systems developers and computer users, and system developers and programmers.

Let's examine an example. The system development process usually begins with written descriptions of the information to be processed. Normally systems developers are expected to describe tasks, information flow and algorithyms for processing the information. In an accounting system, to take an example, system analysts would identify the formal documents that are routinely processed, list the tasks that people would perform with these documents and then write-up mathematical formulas for calculating the necessary budget entries. All logical steps that help insure that the bills get sent out and the accounts recorded. But what part of reality do such descriptions describe? System specifications then develop these descriptions into even more concrete formulas that are used for translating the problem into specific hardware and software requirements. In the process of writing descriptions and specifications many questions get pushed to the side. Questions and issues that don't fall in the realm of objective, impersonal quantification.

When system developers look at a workplace they are more likely to follow the trail of information than to *observe* what is actually taking place. Issues such as--where people sit, whether or not they can talk to each other as they work, how they organize their physical environment, are simply not part of the normal system description process. The tools and techniques of system development rely on identifying quantifiable phenomena rather than techniques of *observation.* And this, in turn, affects the work process, for even when programmers and analysts *are* working together they are still *dividing* the world that they see into objective parts. What categories of information are used? How often is the data retrieved? Who has access to the data? These questions are the stock-in-trade of system developers. As they cut out the patterns for a new system they simply cut away the parts that don't fit into formal descriptions. In so doing, they are not only separating *emotional from intellectual labor*, but also separating themselves from the people who will use the system

Patterns of Communication

Methods and work processes are firmly grounded in patterns of communication and use of *language*. Different groups of workers use different languages and play language games according to their own rules (13). For the systems analyst the *written word* is golden since it forms the key part of the description and specification process. Computer code plays the same role for the programmer. And spoken language and actions are usually most important among groups of computer users, like office or production workers .

Lack of awareness of the language games, results in further divisions in work organization. Often, for example, systems analysts rely on written job descriptions to try to understand what users are doing. But the differences between the job description and the actual *tacit knowledge* used on the job, is quite large. Many analysts complain that when they try to get users to describe what they actually do, they can't understand what the users are saying.

These problems of language games are in no way unique to the systems development process. But as long as the organization of systems work continues to ignore these problems, language will loom large in continuing to divide labor. Non-verbal cues, as well as behavior and attitudes express many key elements in the way computer systems may be used. Yet these forms of expression remain burried beneath the language barrier that sits as a wall between objective and subjective techniques. Behavior, attitudes and emotions stay on the 'female' side of the wall.

And there is a still, more complex problem involved in language games that effects the way groups and individuals relate to one another. These more subtle differences are expressed in the way the language games are played. The use of argument and debate, for example, is a well used technique among scientists, academics and computer people. Systems developers are expected to express their ideas in concrete, logical terms. Arguments are expressed, counter-arguments posed, and debate is expected. One can argue by saying 'I think', for example, but not begin a debate by saying 'I *feel'*. The expected *style of discussion* is built on the norms and rules of objective science. Discussions about system descriptions usually follow these forms. Systems developers are expected to 'stand up for' their ideas in the arena of debate.

Again, there is nothing, per se, wrong with the impersonal, rational style of discussion. Its time-tested use sets the rules for discussion and agreement. Yet it's a well-known fact that men are socialized to better 'stand up' in this tradition than women. Given the rules of the game, women must park their socialized emotions outside the door of the meeting room.

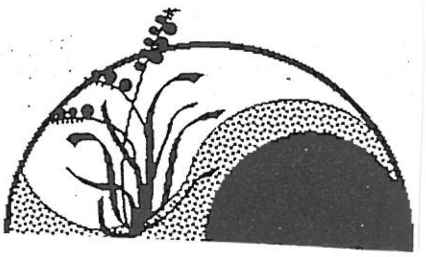
But the differences in discussion style do more than highlight socialization patterns for men and women. The rules of debate also sweep under the rug the personal, non-verbal, more intuitive expressions that can find no voice in this form of exchange. In doing so, they further define and cut away pieces of reality that are needed to better understand how computer systems might be used. Like the language barrier, these rules separate systems developers from users, for computer users are rarely schooled in these forms of argument.

Imagine, if you will, a discussion between a group of clerical workers, who as computer users, have some concern about the introduction of a new computer system. The clerical workers, despite all of their tacit knowledge about their jobs, are little match for system developers in the 'hard-line' negotiations about what the new system will do. And the clerical workers concerns stand behind another line of debate--that between management's objectives for the system and the way the workers want to do their job. If the system developer gives logical arguments for the new system, and the user *feels* that something is wrong; there can be no discussion. The systems developers' methods, language and style are labelled as 'professional good practice', the users ideas may be called 'vague' or 'emotional'. Given the rules of the game, user objections get treated as 'less important'.

For women workers the problems are particularly accute, for socialization twarts their ability to 'play hard ball' (as negotiations are frequently called) with managers and system developers. But as long as the language games of "technical rationality" reign supreme, the problems continue for all groups of computer users. Recently, a navigator on a large passenger ship told this story. A computer system was introduced on-board to log-in all entries the navigators and technical assistants made during the course of a voyage. The navigators had been used to keeping all information in front of them--all written in the time-tested way of journal entries in the ship's book. The computer system was rather awkard to use. Each time the navigators had to find out what was going on they had to enter their code into the computer terminal and 'call up' a specific piece of information. Not only did the system slow down their normal work practices, but it critically interrupted the traditional routines. It was much easier for navigators to simply talk to each other with the ship's book in front of them, then for each person to log-in their information in the isolated environment of the computer terminal.

Most practicing system developers would label this story as an example of 'bad' system design. Yet the issue remains that 'bad' systems seem to be more the rule than the exception. Of course systems developers can learn a great deal by borrowing methods from the social sciences and the humanities. And use of economic, political and philosophical analysis can further the cause of 'good system design'. But we must keep in mind, that these other disciplines (sometimes labelled as 'soft sciences'!) come marked by the same gender dichotomies that frame the systems development process. Evelyn Fox Keller reminds us that "both gender and science are socially constructed categories". Clearly, the development and use of computer systems are socially constructed. In our study of how computer systems are socially constructed we need to dig more deeply to further uncover the roots of gender bias in the methods, work organization and languages in use. And gender analysis can give us a set of glasses to better view some of these problems, for it focuses on the issues that are cut from the heart of the matter.

Gender analysis, while useful, should also be viewed within the context of the economic systems that it supports. The roots of what is considered 'good science' are deeply planted in the period that gave rise to economic expansion in the form of early capitalism. In sweeping away traditions and other 'less objective' customs, the economic motions of capitalism have used gender dichotomies to stay afloat. Indeed the division between the 'head and the heart' remains as a central issue in both the way computer systems are developed and in their economic and political uses.



To end on a personal note, I began this story with a reference to the my experiences as a system developer. I love systems development work - it is an exciting feeling to create something, as if, from nothing. I always felt happiest when deeply involved in the creative, imaginative work of designing new systems. Imagining the final system and fantasizing about how it might work is almost as thrilling and terrifying as being pregnant. But with each system that I worked on, I always felt that a part of myself was missing. Each time I had to describe a work process or argue the logical necessity of the chosen 'solution', I had to quiet my heart and prove my worth with only my mind. I did it, and I was considered successful. But there was a price. Everyone, I think, pays this price. Men may be socialized to ignore it more often, but I have a feeling that the effect is similar.

The Head and the Heart

End Notes

1. This paper began its life as a working paper following nine months of discussion between Randi Markussen, Gitte Møldrup and myself (Joan Greenbaum). During the gestation period of this paper, our discussions ranged from our reading on gender and science issues, to the preparation of our courses in system development, and, of course, like most processes, included our own feelings and personal experiences about the separation of the *head and the heart* . Given the time pressures at the end of the academic term, and the difficulties of the English language, this version of the paper has been written by me. As such, it is based on my life as a system developer, and while I am deeply indebted to the process of collective discussion, the story told in this version reflects my own experiences.. The collective hopes to continue our working discussions, and, with the help of readers' comments, plans to write further articles on this difficult subject.

2. Eveyln Fox Keller,Reflections on Gender and Science,(Yale Press, 1985). pp.6-7.

3. Harry Braverman, Labor and Monopoly Capital, The Degradation of Work in the Twentieth Century,(Monthly Review Press, 1975); and see Joan Greenbaum,In the Name of Efficiency,(Temple University Press, 1979).

4. see for example, T. Winograd and F. Flores,Understanding Computers and Cognition,(Ablex Publishing, Norwood, N. J.,1986); and Susanne Bødker,*Through the Interface--a Human Activity Approach to User Interface Design,* Computer Science Dept, Aarhus University, 1987.

5. Fox Keller,Op. Cit.,p.12.

6. Ibid. p.7.

7. Ibid., p.11.

8 see, for example, system textbooks such as: William Davis,Systems Analysis and Design, a Structured Approach,(Addison-Wesley, 1983), and James Senn,Analysis and Design of Information Systems,(McGraw-Hill, 1984); and the works of Herbert Simon such

as The Shape of Automation for Men and Mangement,(Harper and Row, 1965).

9. see for example, Eleanor Wynn,*Office Conversation as an Information Media,* Department of Anthropology, Univeristy of California, Berkely, 1979; and Peter Bøgh Andersen,"Semiotics and Informatics: Computers as Media, in P. Ingwersen, et. al Information Technology and Information use (Taylor Graham, London, 1986).

10. Gro Bjerkness, Pelle Ehn and Morten Kyng, Computers and Democracy--A Scandinavian Challenge(Gower, England, 1987).

11. Greenbaum, Op.Cit.

12. see Joan Greenbaum, Andrew Friedman,"Wanted Renaissance People", Datamation,Sept. 1, 1984, and **ICON** (International Computer Occupations Network)**Working Papers,** Bristol University, Economics Department, England, 1985-7; and Sarah Kuhn, "From the Back Office to the Front Lines: The Computer Software Development Labor Process in a Chainging Business Environment", MIT, Department of Urban Studies and Planning, May, 1987.

13. L. Wittgenstein, Philosophical Investigations (Blackwell, Oxford, 1953); see also Pelle Ehn *Human Centered Design and Computer Artifacts,* Aarhus University, 1987, in preparation.