Title
Computing as social action : the social dynamics of computing in complex organizations

Authors
Kling, Rob
Scacchi, Walt

Publication Date
1979-11-07

Peer reviewed

Computer Science

COMPUTING AS SOCIAL ACTION:
THE SOCIAL DYNAMICS OF COMPUTING
IN COMPLEX ORGANIZATIONS

by

Rob Kling and Walt Scacchi*

Department of Information and Computer Science
and
Public Policy Research Organization
University of California, Irvine
Irvine, CA 92717

November 7, 1979

TR 142

TECHNICAL REPORT

UNIVERSITY OF CALIFORNIA
IRVINE

COMPUTING AS SOCIAL ACTION: THE SOCIAL DYNAMICS OF COMPUTING IN COMPLEX ORGANIZATIONS

by

Rob Kling and Walt Scacchi*

Department of Information and Computer Science
and
Public Policy Research Organization
University of California, Irvine
Irvine, CA 92717

November 7, 1979

TR 142

To appear in:
Advances in Computers Vol. 19. Editor, Marshal Yovets
Academic Press, New York, 1980

This work was supported by NSF Grant MCS 77-20831. The analyses reported here were sharpened by discussions with colleagues at U.C. Irvine and elsewhere. In particular we wish to thank James Danizger, William Dutton, Leslie Gasser, John Leslie King, Kenneth Kraemer and Jim Neighbors. In addition, Peter Freeman and Elihu Gerson provided helpful and careful reading of an early draft of the manuscript.

1.0 PERSPECTIVES ON COMPUTING IN ORGANIZATIONS

1.1 The Social Side Of Computing In Organizations

During the last 30 years, computer technologies have evolved from unusual laboratory instruments into common fixtures. Most businesses that gross over $100 million per year in sales, and government agencies that serve publics of several hundred thousand, have automated some of their operations. Computer technologies are becoming widespread and increasingly visible in small businesses and even homes. One can easily count the increasing number and variety of applications which have been automated. What these technologies mean for the people and organizations who use them is much less clear and less amenable to explication by simple measures like dollars spent and dollars saved (Gottlieb and Borodin, 1973).

Almost anybody who deals with computer use has occassion to explain why some computing arrangements work out the way they do. A company president may wonder why the staff of several operating divsions each request their own minicomputers. A systems analyst may wonder why the prospective users of a new information system continually change their minds about their requirements. A bank auditor may wonder whether he will have a harder or easier time understanding the records that are placed on a computer that uses a data-base management system. City councilmen may wonder whether use of a "fiscal impact model" by planners will facilitate understanding of the costs and revenues associated with new commercial and residential developments. The customer of a credit card firm may wonder why a billing error is still uncorrected after several phone calls and promises made over a three-month period.

Questions like these are commonplace. Our answers are usually based upon some assumptions about how the particular computing technology works and how people in the organization who develop, provide, or use computer-based services go about their business. Technical assumptions include simple understandings such as "Data stored in a common data base should be identical for each user." But one might even have to invoke more complex understandings of a particular application in order to appreciate why one's airline reservations for a whole trip are cancelled when only one leg of the trip is cancelled. Similarly, simple and subtle assumptions must be made about the behavior of organizations and their participants. Under what conditions do lower level staff usually follow policies set by higher level managers? Will the activities of groups outside of the computer-using organization affect its computer operations? When can Federal funders, government or private auditors, vendors, competitors, or professional associations be ignored and when are they important parties? Should one view complex organizations largely as composed of groups working toward a common goal, or are they better understood as federations of groups which build a shifting set of alliances and in the process serve selected interests, legitimate and illegitimate? In order to give good answers to these questions, an analyst must understand computing technology and life in organizations sufficiently to model the situtation being examined.

We believe that the technical aspects of most computer systems in use today are relatively well understood. In contrast, the behavioral and social aspects of computing are poorly understood. This chapter examines the ways in which the behavior of people and groups in organizations influences the development, use, and consequences of computing. It indicates recent research findings which help answer questions like those listed above.

When one asks why organizations adopt computing or why some people are more enthusiastic about particular applications, different analysts provide different answers. Some analyses emphasize the economic efficiencies provided by computing (Kanter, 1977). Other analyses emphasize the ways in which automated data analysis provides users with important political resources (Laudon, 1974; Kling, 1978b). Still other analyses emphasize the ways in which state-of-the-art computing use enables both computer specialists and users to enhance their image within social worlds that value innovative activities (Kling and Gerson, 1977). Should questions about the use or consequences of computing in organizations be answered primarily in terms of rational economic calculations, interpersonal relations of key staff, routinized organizational procedures, or the politics of the computer-using organizations? Each of these phrases indicates a different perspective from that the social activities that constitute computing development and use in organizations can be understood. In this chapter, we shall examine six common perspectives that provide different terms or "storylines" with which to understand how people live and work with computing in organizations. This chapter can be read as a review of recent studies of computing development, use, and impacts in organizations. It can also be read as an examination of the usefulness of different social perspectives for explaining how computing developments "work" in complex organizations.

1.2 The Case Of A Client-tracking Welfare Information System

These six perspectives are best introduced by indicating how they help explain a complex case of computer use. First we will introduce the case of a welfare information system used by the staff of an American local government. Then we will indicate how analysts, using each of six different perspectives on behavior in organizations, would view this case.

In the mid-sixties, the U.S. Department of Health, Education, and Welfare funded a major municipality, "Riverville," to build an automated information system (UMIS) to help "integrate services" among dozens of different local welfare agencies. This integration would help improve administrative efficiency as well as the quality of service provided to welfare clients in Riverville. Agreements to this effect were given by municipal staff and representatives of the computer vendor, who jointly developed the earliest, batch version of UMIS.

Data provided by welfare clients upon application for welfare assistance were entered on UMIS at one of several information and referral offices supported by public funds. Participating agencies were also requested to enroll their clients on UMIS. Whenever a person was referred to a service, it was recorded on UMIS. Whenever the services were provided, that too was recorded on UMIS.

Staff in the welfare agencies were suspicious of UMIS. A cable connecting UMIS to one of the neighborhood service centers was cut just as it was being introduced. Through administrative fiat and persuasion, UMIS was adopted by the staff in the municipally operated neighborhood service centers and 34 welfare programs. The staff of other agencies that had small clientele or that were large and had their own automated recordkeeping systems were reluctant to utilize UMIS.

By 1970, the technology of both UMIS and of welfare operations had changed. UMIS supported sets of terminals in several neighborhood centers to ease the entry and retrieval of data about individual clients or their families. By 1970, 170 different local agencies were listed in a directory of "participating agencies" published by the staff of UMIS. Also, the 34 welfare programs operated by the city were administratively consolidated into one agency, and UMIS was supposed to help evaluate programs and eliminate duplicate paperwork. A careful study conducted in the mid-seventies showed that UMIS was in place and used routinely for recording the transactions between some agencies in Riverville and their clientele. However, data on the services provided to many clients was incomplete because many agencies refused to use UMIS. Data was also inaccurate since UMIS was used as a training aid for a job training program designed to teach people data entry skills. UMIS provided little useful information to welfare caseworkers and it had minimal influence on their work styles. UMIS did not help integrate the operations of any agencies and was not used to increase administrative efficiency. Rather, its reports were used to help generate more Federal funding by taking advantage of the fact that HEW auditors believe an agency with such extensive automation must be well managed, and that computerized data is more credible than manually aggregated data [Kling, 1978b].

1.3 Six Perspectives On Computing In Organizations

When confronted with a complex set of events involving many participants over 10 years, we are easily overwhelmed. How do we focus our attention and make sense of the bewildering array of facts and relations which appear in the case of UMIS? When people knowledgeable about computing in organizations attempt to explain UMIS operations, effects, and failures, they usually adopt one of six perspectives:

1. *Rational* analyses emphasize the formal ends which computer-based technologies are supposed to serve. Rational analysts emphasize economic payoffs of computing, and explain its value, diffusion, successes, and failures by the ways computer applications help increase the information processing capacity of social actors, and thereby help them satisfy their espoused goals. Rational analysts explain the discrepancies between the espoused goals of UMIS and its actual patterns of use by indicating that UMIS was based upon a simple technology (Hiltz and Turoff, 1978). Hiltz and Turoff argue, for example, that UMIS would "really" have helped caseworkers if only they had utilized computer conferencing.

2. *Structural analysts* share with rational analysts an emphasis upon the formal tasks claimed to be important by the most credible officials in an organization. Structural analysts enrich the rational account by situating decisions in the context of ongoing organizational activites which may be scattered across organizational subunits, episodic, and disrupted repeatedly by alterations in the environment. Structural analysts have typically viewed the the choice of the "best" information system for an organization as contingent upon some features of the organization's activities and the stability of its environment (Galbraith, 1977). Structural analysts are concerned with the ways in which "structural properties" of organization such as size, complexity, and centralization are influenced by information systems. For structural analysts, the key questions about UMIS would be whether it helped top administrators consolidate several welfare agencies into one super-agency and whether it helped centralize decision-making in their hands. They would also be interested in the reasons why UMIS did not help integrate the operations of the diverse array of welfare agencies situated in Riverville.

3. human relations analysts are particularly concerned that the people using an automated system are not constricted by its use. This perspective, commonly adopted by organizational psychologists, focuses upon the way in which different computing arrangements alter the quality of working life of participants. Concerns with job satisfaction, motivation, and alienation are focal for these analysts. When faced with UMIS in Riverville, human relations analysts would wonder why some people cut the cable between UMIS and a neighborhood service center. They would be interested in whether the shift from a batch system to an on-line operation eased the time pressures and demands for accuracy faced by clerks who were entering and retriving clients records. They would wonder whether UMIS was used to monitor the workloads of caseworkers, and to increase closeness of supervision.

4. Interactionist analysts view computing technologies as objects which can take on rich social meanings for people who deal with them. Analysts of this persuasion are concerned with the ways in which people define their social situations, and the ways in which they create strategies of action in line with their perceptions and intentions. Interactionist theorists believe that many of the rules and policies which are supposed to regulate life in organizations are in fact moderately flexible, fluid, and frequently altered, through interpersonal negotiations, to fit the lives of the participants. Interactionist analysts who studied Riverville would focus on the conceptions of UMIS held by different participants in the local welfare agencies, the municipal administration, and the local computing world. They would wonder why HEW auditors believed that data from a computer were more credible than manually counted data. And they would be interested in learning how municipal managers developed UMIS as a negotiating instrument to deal with Federal auditors. They would also ask whether the relations between clients and

	RATIONAL	STRUCTURAL	HUMAN RELATIONS	INTERACTIONIST	ORGANIZATIONAL POLITICS	CLASS POLITICS
TECHNOLOGY	Equipment as instrument	Equipment as instrument	Equipment as instrument/environment	"Package" as milieu	Equipment as instrument	Equipment as instrument
SOCIAL SETTING	Unified organization 1. The user 2. Tasks and goals 3. Consistency and consensus over goals (assumed)	Organizations and formal units (e.g., departments) 1. Formal organizational arrangements 2. Hierarchy of authority, reporting relationships	Small groups and individuals 1. Task groups and their interactions 2. Individual needs 3. Organizational resources and rewards	Situated social actors 1. Differentiated work organizations and their clientele 2. Groups with overlapping and shifting interests	Social actors in positions 1. Individuals/ groups and their interests	Social classes in stratified system
ORGANIZING CONCEPTS	* Rationalization * Formal procedures * Individual ("personality") differences * Intended effects (assumed) * Authority * Productivity * Need * Cost-Benefit * Efficiency * Task * Better Management	* Organizational structure * Organizational environment * Communication * Standard operating procedures * Organizations resources and rewards * Uncertainty absorption * Rules * Authority/power * Information flow	* Trust * Motivation * Expectations and rewards * Job satisfaction (subjective alienation) * Self-esteem * Leadership * Sense of Competence * User Involvement * Group Autonomy	* Defining situations * Labeling events as a social construction * Work opportunities/ constraints * Package * Career * Legitimacy * Social world * Social conflict * Interaction * Role * Negotiations	* Work opportunities/ constraints * Power * Social conflict * Legitimacy * Elites * Coalitions * Political resources * Bargaining * Power Reinforcement	* Ownership of means of production * Power * Social conflict * Alienation * Deskilling * Surplus Value
DYNAMICS OF TECHNICAL DIFFUSION	1. Economic substitution--"Meet a need" 2. Educate users 3. A good technology "sells itself"	The fit between attributes of the innovation, organization, and environment enable diffusion	Acceptance through participation in design	Accepted technologies preserve important social meanings of dominant actors	Accepted technologies serve specific interests	Accepted technologies serve dominant class interests
GOOD TECHNOLOGY	1. Effective in meeting explicit goals or "sophisticate" use 2. Efficient 3. Correct	Helps organizations adapt to their environments	Promotes job satisfaction (e.g. enlarges jobs) and improves job performance	Does not destroy social meanings important to lower level participants, public and underdogs	Serves the interests of all legitimate parties and does not undermine legitimate political process	1. Does not alienate workers 2. Does not reproduce relations
WORKPLACE IDEOLOGY	Scientific management	Scientific management	Individual fulfillment through work	Individual fulfillment through evocation of valued social meanings	[Several conflicting ideologies]	Worker's control over production

TABLE 1

SIX PERSPECTIVES ANALYZING COMPUTING

welfare agencies were altered by virtue of UMIS. Were welfare applicants aware of its presence? Did it alter their perceptions of how formal or helpful the agency staff was?

5. Some analysts emphasize the political character of organizational life. Participants in organizations are viewed as players in positions from which they can command certain resources, and will often jockey to increase their influence. They view the decisions made by organizations as the actions in power games, or as the resultants of many such actions. An organizational politics analyst would select the power dimensions of social life for careful scrutiny. Like the structural analyst, he would be concerned with whether top administrators in Riverville were able to use UMIS to enhance their control over welfare operations. Like the human relations analyst, he would be interested in the extent to which casework supervisors used UMIS to increase their control over the activities of caseworkers and clients. He would also focus on the relations of power and dependency between the different welfare agencies in Riverville. Last, he would highlight UMIS' utility as an instrument of bureaucratic politics between local officials and federal auditors and funders.

6. Class politics analyses may be viewed as a special kind of political analysis. Usually using Karl Marx as a point of departure, class analysts examine how the use of computing in organizations reinforces existing class relationships. In particular, they are concerned that workers become increasingly disenfranchized and work becomes increasingly degraded in automated workplaces. Ironically, class analysts often share the assumption of rational analysts that automated information systems cannot help but effectively rationalize the operations of computer using organizations. Class analysis would explain how the American welfare system is designed to foster a labor pool

	RATIONAL	STRUCTURAL	HUMAN RELATIONS	INTERACTIONIST	ORGANIZATIONAL POLITICS	CLASS POLITICS
TECHNOLOGY	Equipment as instrument	Equipment as instrument	Equipment as instrument/environment	"Package" as milieu	Equipment as instrument	Equipment as instrument
SOCIAL SETTING	Unified organization 1. The user 2. Tasks and goals 3. Consistency and consensus over goals (assumed)	Organizations and formal units (e.g., departments) 1. Formal organizational arrangements 2. Hierarchy of authority, reporting relationships	Small groups and individuals 1. Task groups and their interactions 2. Individual needs 3. Organizational resources and rewards	Situated social actors 1. Differentiated work organizations and their clientele 2. Groups with overlapping and shifting interests	Social actors in positions 1. Individuals/groups and their interests	Social classes in stratified system
ORGANIZING CONCEPTS	* Rationalization * Formal procedures * Individual ("personality") differences * Intended effects (assumed) * Authority * Productivity * Need * Cost-Benefit * Efficiency * Task * Better Management	* Organizational structure * Organizational environment * Communication * Standard operating procedures * Organizations resources and rewards * Uncertainty absorption * Rules * Authority/power * Information flow	* Trust * Motivation * Expectations and rewards * Job satisfaction (subjective alienation) * Self-esteem * Leadership * Sense of Competence * User Involvement * Group Autonomy	* Defining situations * Labeling events as a social construction * Work opportunities/constraints * Package * Career * Legitimacy * Social world * Social conflict * Interaction * Role * Negotiations	* Work opportunities/constraints * Power * Social conflict * Legitimacy * Elites * Coalitions * Political resources * Bargaining * Power Reinforcement	* Ownership of means of production * Power * Social conflict * Alienation * Deskilling * Surplus Value
DYNAMICS OF TECHNICAL DIFFUSION	1. Economic substitution--"Meet a need" 2. Educate users 3. A good technology "sells itself"	The fit between attributes of the innovation, organization, and environment enable diffusion	Acceptance through participation in design	Accepted technologies preserve important social meanings of dominant actors	Accepted technologies serve specific interests	Accepted technologies serve dominant class interests
GOOD TECHNOLOGY	1. Effective in meeting explicit goals or "sophisticate" use 2. Efficient 3. Correct	Helps organizations adapt to their environments	Promotes job satisfaction (e.g. enlarges jobs) and improves job performance	Does not destroy social meanings important to lower level participants, public and underdogs	Serves the interests of all legitimate parties and does not undermine legitimate political process	1. Does not alienate workers 2. Does not reproduce relations
WORKPLACE IDEOLOGY	Scientific management	Scientific management	Individual fulfillment through work	Individual fulfillment through evocation of valued social meanings	[Several conflicting ideologies]	Worker's control over production

TABLE 1

SIX PERSPECTIVES ANALYZING COMPUTING

of low paid workers who can be easily exploited by potential employers. They would argue that UMIS does nothing to alter the fundamental relations between social classes within which welfare services are provided. They would also expect UMIS to lower the quality of working conditions of staff in the welfare agencies. In particular, they would share with human relations and organizational politics analysts a focus on the role of UMIS in altering the power relations between workers and their managers.

Most of the articles and books which explain some facet of computing in organizations are primarily informed by one or another of these six perspectives.[1] None of these approaches is unique to computing. Analyses representing each approach may be easily found which examine other aspects of technology and organizational life. Each perspective provides a way for the analyst to focus on a few relations of particular interest, and to develop a coherent storyline about his topic. In Table 1, we indicate some of the key features of each perspective.

[Place Table #1 about here]

Each perspective includes a view of technology and a way of defining the social setting in which the technology is used. For example, both rational (Hiltz and Turoff, 1978) and organizational politics (Laudon, 1974) analysts view computing as an "instrument" or tool which is directed to some specified end. In contrast, the interactionist views computing as creating a complex social milieu which brings a variety of participants, including "users," into a rich, shifting, and sometimes problematic set of social arrangements (Kling and

Gerson, 1978; Kling and Scacchi, 1979). Rational analysts typically identify the individual or group who uses computing for the major focus of attention. Analysts who adopt an interactionist (Kling and Gerson, 1977, 1978) or political perspective (Laudon, 1974; Kling, 1978b) identify a variety of groups and individuals who participate in the social and organizational world of the identified user as critical actors.

Analyses written from a particular prespective are framed with some of the <u>organizing concepts</u> identified in Table 1. An analysis of UMIS in Riverville which centrally emphasizes the rationalization of administrative procedures and its intended effects, should be understood as a rational analysis. Rational analyses typically indicate the utility of computing to some individual or group with little attention to variations between them or their organizational worlds. Similarly, rational analysis often slight examination of different modes of computing and their fit in a variety of settings. If the conceptual vocabulary is enriched to account for different organizational arrangements of computing development, operations, and use (e.g., centralization of control), or if the analyst pays attention to the critical uncertainties in resources and lines of action which actors cope with in an organization, then one is reading a structural analysis. In practice, analysts may draw upon the organizing concepts of two or more perspectives. However, some couplings are most common. Some structural analyses reflect the arid, consensual world of the rational analyst (Galbraith, 1977). Some rational analysts indicate a few structural features of the computing world they examine (e.g., transaction volumes, or organizational hierarchy) without explicitly hinging the validity of their analyses to carefully specified structural conditions (Hiltz and Turoff, 1978). The frameworks of interactionist and political analysis are sometimes blended, particularly in the studies of the symbolic politics of computing and social power (Kling, 1974, 1978a; Kling and Gerson, 1977). Couplings of political and structural anlsyses are also often common

(Greenberger, et al, 1976;Colton, 1978, Goodman, 1979; Kraemer, et al, forthcoming.) But most analyses emphasize one dominant set of organizing concepts and selectively utilize others to fill in conceptual gaps on an ad-hoc basis.

In this chapter, we will indicate how analyses of computing system developments, uses, and consequences for organizational life have been informed by these six alternative perspectives. The major value of identifying these perspectives is not to provide a static typology for sorting studies. Rather, it is to indicate how critical assumptions about the social world pervade common arguments about the development, use, and consequences of computing. These six perspectives indicate some of the major clusters of social assumptions and organizing concepts that inform contemporary analysts. The concepts of each perspective have been developed and used in relatively rich literatures devoted to both theory and application. Because of space limitations, we will not elaborate the core concepts of each perspective that are listed in Table 1. Each perspective has a rich and distinct literature. The interested reader may find detailed examinations of these perspectives in Allison (1971), Perrow (1979), Cuff and Payne (1979), and Braverman (1974).

We will examine software development in different stages of the system life cycle, some elements of computer use, and the impacts of computing on decisions, power, and worklife in organizations. There are many topics we are not examining. We have not selected particular technologies such as minicomputers or decision-support systems for scrutiny. Nor have we examined certain managerial schemes such as radically (de)centralizing computer operations, appointing policy boards, or full chargeback pricing. The computing world is alive with many novel and interesting technologies. There are many strategies for most effectively developing them and managing their use. Rather than examining these options in fine detail, we examine the efficacy of the six perspectives commonly used to analyze new technologies and the social arrangements for managing their operation.

How computing "really" fits into the social life of organizations depends in part upon one's particular theoretical preferences. Structural analysts "see" a world of communication channels, standard operating procedures and environmental uncertainty (Galbraith, 1977). Interactionists "see" a world of people negotiating their own definitions of potentially fluid situations. These social meanings are constructed and pursued through interactions with others and with technologies at hand. These social interactions, and their focal concerns related to computing, will vary in place, time, and context (Kling and Gerson, 1977, 1978). Political analysts, in contrast, "see" a world in participants actively play and manipulate to increase their power (Kraemer and Dutton, 1977). While these perspectives denote languages which punctuate what will be seen and what will be ignored, they do not determine the substantive beliefs of an analyst about how computing "works" in different settings.

Within any given theoretical perspective there can be serious debates or marked consensus over "what is really happening." Rational analysts generally agree that management information systems have been oversold and "underused" by managers. Recently, two researchers tested the viability of the theoretical assumptions of rational and political perspectives within the same study (Kling, 1978b; Markus, 1979) and found that rational models explained a much narrower range of important behavior. However, there is no consensus among analysts of organizational politics whether automated information systems enhance the power of the topmost managers in an organization (elite politcs), or simply enhance the power of those already strongest in an organization (reinforcement politics). For example, in large organizations, such as the U.S. Federal government or horizontally merged conglomerates, heads of the largest departments or operating divisions, not the president, may exert effective influence over what business is done and how it is done.

We are not indifferent in our choice of perspectives. We believe that the interactionist and organizational politics perspectives go farther in providing appropraite languages and conceptions of social dynamics to help explain the attractions and dilemmas of computer use in organizations than do their competitors. However, we believe the social elements of computer use will be understood much more keenly if we explicitly contrast analyses from different perspectives.

Analysts also differ with respect to the _unit of analysis_ they focus upon. Some analysts emphasize the behavior of participants _in_ organizations, others focus on the behavior _of_ organizations, and still others try to link the two (Kling, 1978b). For example, a structural analyst might claim that organizations will adopt computing to help increase the ease of coordinating large volumes of routine transactions. However, an analyst who studies the specific computing arrangements used by the staff of such a firm may find its economic rationality less obvious and would emphasize the politics of its computing decisions (Pettigrew, 1973). (He might have found that the board of directors of a firm dominate decisions on choice on equipment and choose equipment from a prestigious, but expensive vendor rather than the more flexible and technically appropriate equipment of a competitor.) These two accounts are congruent even though they are couched in different theoretical terms. In this case the structural analyst is describing the _behavior of the organization_ while the political analyst is scrutinizing the _behavior of its participants_. A political account which truly conflicts with this structural account would focus on the behavior of the _organization_, rather than just of its _individual members_. In contrast, an analyst who argues that the staff of an organization adopted computing to enhance their image of being efficient or progressive, rather than to enhance their ability to coordinate routine operations, would be a contending analysis from a political perspective.

In this chapter, we will examine these different social elements of computing development and use in organizations. We will focus upon what we have learned about the actual patterns of computer development and use in "representative" organizations, and emphasize the findings of careful empirical studies of computing in organizational life. We believe that careful studies of computing development and use in real organizations offer an essential set of data to help us understand how computing arrangements of newer and older design may "work" in organizations, bearing in mind that the "empirical" nature of studies does not mean that they look for the same things in studying the social aspects of computing, or that they interpret the findings in the same way.

Typically, the social aspects of computing development and computing use are treated separately. Some management analysts and software engineers have been concerned about ways to organize software development and computer operations so that they are smooth, cost effective, and efficient (Cooper, 1978). Other analysts have focused their attention on the social repercussions of computer use (e.g., worklife, organizational structure). We view these diverse topics as pieces of a common fabric. Whenever computing is embedded in an organization, it becomes a markedly social phenomenon. This viewpoint is ironically underlined by the complaints of people who adopt a rational perspective and who wish that people or politics could be taken out of computing. "Supportive" managers, "clever" programmers, "indifferent" machine operators, "career-oriented" auditors, "stupid" users, and "foot-dragging" vendors often appear on the playbill of the rational analysts who wish that systems could be perfect in their theater, and that the actors would depart, leaving the director alone with the plans, technology, and the audience (Strassman, 1973). But with such a cast of characters on hand, something of interest might just be happening; and often it is.

In the following sections, the development, use, and impacts of computing in organizations will be examined in light of the six perspectives outlined above. Section 2 examines the development and provision of computer services through the life cycle from initiation to evaluation. Section 3 indicates how knowledge about computing is distributed throughout organizations and how this leads to systematic misperceptions of computer use and increases the likelihood of computing errors. Section 4 examines the consequences of computer use for the ways decisions are made, the worklives of computer users, and the distributions of power in computer-using organizations.

The narrative thread we have woven is at times, somewhat complicated as we examine topics such as software maintenance, discretion in computing use, and the role of computing in decision-making from several different perspectives. Usually, each topic has been most carefully examined by analysts who represent only a few of the six perspectives. In principal, we would wish to examine each topic from each perspective, and provide a more consistent development. Two difficulties intrude. Primarily, the literature composed of careful studies of computing have extremely fragmented, and researchers have typically not developed lines of inquiry that examine these diverse topics from any one theoretical perspective. Sometimes we have filled in appropriate analyses when we believed them to be particularly relevant and we could ground them in data collected during the course of our own fieldwork. Otherwise, our gaps reflect those in the literature. For example, human relations and class politics (Kraft, 1977) analyses of computing work in general can be developed straightforwardly. But, peculiarly human relations or class politics approaches to software testing are less transparent. In Section 5, we integrate the major themes, and suggest lines for further work.

2.0 THE COMPUTING SYSTEM LIFE CYCLE

Some analysts conceptualize computing systems as developing through a temporal sequence of stages, a "life cycle." A computer system is designed before it is implemented, tested after it is implemented, and maintained until it "dies." Organizing research by phases of the system life cycle focuses attention upon the generic activities of system development integrally linked with computing use (cf. Learmonth and Merten, 1978). Traditionally, the computing system life cycle draws attention to the different phases of system development independent of their organizational setting (Cave and Salisbury, 1978; Cooper, 1978). This rational tradition reflects the desire by many analysts and computing specialists to view computing system development, use, and maintenance as essentially technical and mangerial endeavors (De Roze and Nyman, 1976). But the tide may be beginning to turn. For example, informed software engineers have begun to see that

> "concerns for the <u>social implications of computer systems</u> are part of the software engineer's job, and techniques for dealing with these concerns must be built into the software engineer's practical methodology, rather than being treated as a separate topic isolated from our day-to-day practice." (Boehm, 1979; emphasis added)

We will examine some of the concerns encountered by computing users, specialists, and managers according to the perspectives that have informed the analysis of different techniques.

In this section, we will show how each phase of the computing system life cycle is shaped predominantly by social interactions between computing promoters, developers, users, maintainers, and their organizations. We will investigate each of the system life cycle phases in table 2 as covered in the literature. When applicable, we will provide examples of

Table 2.

1. Initiation and Adoption - the activities related to system adoption and requirements specification. Getting the organization ready for the arrival of the intended computing system. The common choice is between developing a system in-house or transferring-in a system developed by some other organization.

2. Requirements Specification - the activities which outline the design of a computing system. The system specifications are determined by some mix of managers, computing specialists, and instrumental users.

3. Selection - the activities surrounding the choice of where to acquire a computing system.

4. Design - the activities dealing with the detailed specification of the structure and functions of the computing system. Computing systems can be distinguished according to whether their designs are primarily user-centered or machine-centered.

5. Implementation - the activity of building and installing a specified computing system in an organization. Some management and MIS analysts include system design in this phase.

6. Testing and Validation - the activity of demonstrating, verifying, or "proving" that a system works as expected.

7. Documentation - the activities in which a written record of the functions, structure, and operating instructions of a computing system is produced.

8. Computing System Use - the activities that entail work with a computing system on the part of managers, computing specialists, other professionals, and clerks.

9. Maintenance, Modification, and Conversion - the activities concurrent to the use of a computing system which keep it alive and operational. Now considered to be the most costly phase of the system life cycle.

10. Evaluation - measuring the fit and performance of a computing system with respect to computing arrangements and organizational goals.

situations drawn from our own case studies. However, we will investigate computer system use separately in section 3.

Many authors and organizations have defined the phases of the computing system life cycle. However, their definitions vary widely with respect to the number of phases, their names, and the activities within each. For this chapter, we define the phases as follows:

Place Table 2 about here

The "system life cycle" as used here does not refer to some predetermined sequence of steps through which all systems must inevitably pass. Rather, it directs our attention to the beginnings, middles, and endings of computing systems that may lead to new systems beginnings.

In organizational settings where computing systems have been in use for some time, new application systems are continually added to the existing system environment. "Systems" and application software tend to be built upon the layers of the existing kernel of software. Application software often depends on systems software for proper operation. As software components grow larger and "bump" into each other, computing systems intended as solutions to certain problems can lead to new problems to which systems must adapt or be replaced. Thus, as a system develops within its life cycle, a number of the "phases" mentioned in Table 2 occur in parallel. One indicator of this concurrency is the continuing growth of software systems found in the use and maintenance phases. That is, once a system is implemented, it either stays in operation and maintenance, or is converted and assimilated within some other system. Most systems are replaced or upgraded upon obsolesence. Few computing systems "die" or disappear without replacement by another computing system.

In the remainder of this section we will review the phases in the system life cycle utilizing the six analytic perspectives listed in section 1. We will then examine a common software system found in modern organizational computing environments, an automated text processing system. Here we will draw upon a recent study we conducted of such a system in use at "CSRO," a computer science research organization (Scacchi, 1980). After presenting this case as a detailed example of a software system life cycle, we will review it with respect to the six analytic perspectives.

2.1 Initiation And Adoption

System initiation is the phase of the system life cycle when a computing system is proposed and the decision to adopt made. In considering computing system initiation, we are interested in the activities surrounding system promotion and the incentives for computing system adoption.

Traditionally, new or better technical components or complete systems are said to move "naturally" from inventor or developer toward users and consumers. Computing system promoters or adopters can, however, be motivated by different, potentially conflicting, incentives. Vendors are interested in selling (or overselling) systems for profit. Federal agencies seek ways to improve their efficiency and spheres of influence. Specialists sometimes want to work with newer, more sophisticated systems instead of procrustean user applications. And users seek ways of improving their productivity or organizational effectiveness with minimal or extensive computing system use.

Rational analysts hold pleasingly simple conceptions of the diffusion of technical innovations. To wit: successful technologies meet a "need" of individuals or organizations (Licklider and Vezza, 1978); as people learn to appreciate that a new technology meets some need better than its alternatives, and as the costs of obtaining the technology decrease, the technology is adopted on a large scale (Simon, 1977).

This rational account is commonly articulated by technologists and the staff of computer-using organizations responsible for developing new computer applications.

The primary analytical difficulty that confronts any scholar who studies these claims arises in the attempt to conceptualize "need." Structural analysts and analysts using an organizational politics perspective approach this problem differently. Structural analysts typically focus on the information processing task to which computing technologies can be applied. Gerson and Koenig (1979) for example, analyze the applicability of computing to different <u>tasks</u> by examining the extent to which the tasks can be easily rationalized. They indicate that organizations which carry out a large volume of easily rationalizable tasks (e.g., insurance companies which process bills, claims, and payments) are much more likely to find computing applicable than are firms where rationalizable tasks are a smaller fraction of the work done (e.g., law firms). Organizational politics analysts, on the other hand, focus on the interests and intentions of participants in a decision to adopt a new computing arrangement.

One important study examined the relative importance of structural and political explanations of computing adoption. Danziger and Dutton (1977) carefully examined the role of institutional needs and the social features within and outside a large sample of American local governments to predict the rates at which they would adopt computing applications. By

Laudon's (1974) insightful case studies of four police computing systems in state and local governments illustrate how these systems served the interests of particular elite groups in the governments which adopted them. Sometimes the introduction of these systems was instrumental in bureaucratic politics. For example, one state governor wished to establish a state attorney general's office, but the local police departments were extremely autonomous and their staff didn't want a state attorney general's office. As a quid pro quo to help establish the attorney general's office, the governor offered to develop a state-run computer system to keep track of wants, warrants, and criminal histories for all jurisdictions within the state. Its administration would be centralized in the new attorney general's office. Thus the governor sought to establish an attorney general's office by offering in exchange a police information network for local police. Laudon views new computer systems as political instruments which are selected to fit the political contours of existing organizations with their own ongoing conflicts and coalitions.

In one of our own case studies of a multinational engineering firm, "WESCO," similar patterns were sometimes found. Chemical engineers at WESCO utilize automated simulations to help calculate the design parameters of chemical processing plants. Sometimes they seek simply more precise estimates for the sizes of pumps, compressors, and costly equipment. But there are also occasions when engineers have trouble convincing auditors hired by their clients of the efficacy of their designs. At times, engineers have moved from hand calculations to available simulation programs in oder to help snow the auditors. The calculations look precise, accurate, and sophisticated. Since WESCO's simulation programs are proprietary, it is also more difficult for auditors to double check the correctness of the calculations or the assumptions made.[2]

In the language of rational analysis, one could say that engineers at WESCO and administrators in Riverville found computing "applicable" to decisions in their organizations (Licklider and Vezza, 1978). But much is lost with such banal characterizations. Sometimes people choose to utilize computer-based systems because they value the expected technical benefits such as more precise calculations, speedier information flows, and more flexibly organized reports. On other occasions, as shown above, computer use serves as an instrument of bureaucratic politics (Kling, 1974; Laudon, 1974). Computing applications used as political instruments do not easily fit the accounts of rational analysts who ignore the politics likely to be found in any organization (Simon, 1973; Burch, Strater, and Grudnitski, 1979). Many analyses of computer use developed within the rational tradition are sociologically naive. These analyses focus on the technical payoffs of computing in organizations and neglect patterns of use that are inconsistent with an economic conception of organizational activity. But to the extent organizational politics influences the move to adopt computing, its dynamics must be part of any systematic understanding of the antecedents or consequences of computer use in organizations.

2.2 Selection

It is commonly held that computer-using organizations benefit by adopting computer applications developed by an outside vendor or some other organization. Computer scientists and federal research supporters have paid serious attention to schemes for developing machine independent software, to help diminish the costs of transferring applications across organizations. Kraemer (1977) studied the actual rates of application software transfer between American cities. He found that a typical city automates about 40 applications, but in general only one has been transferred in. Kraemer and King jointly studied federal projects in which cities were funded specifically to transfer applications, and found that applications were rarely transferred to other cities (Kraemer and King, 1979). One could interpret these findings in a number of ways. One could bemoan the ignorance of public officials and wish that they were "wiser, technically competent, and more innovative." More sensibly, Kraemer analyzed the incentives provided for particular groups in an organization to transfer-in an application. Subsequently, Perry and Kraemer (1978) found that the structural attributes of the organization and of the computing application influenced their adoption and selection.

In most organizations, transferring-in a new application does not provide strong rewards for a local computer group, compared with local development. A manager who supervizes local development can increase or maintain staff, increase his budget, and become an expert on yet another critical operation. Interactionist analysts note that given the development bias of the computing world, a new developer may gain additional prestige compared to an actor who merely transfers-in an existing application (Kling and Gerson, 1977). Together, these create a mobilization of bias against transferring-in new applications.

When computing applications are adopted to help the staff of an organization cope with some information processing demands, there may be a "political" dimension to their decisions. Often, participants in an organization differ over the forms of computing they prefer. Pettigrew (1973) found that the selection of new computer systems made by the board of directors in a British manufacturing firm was influenced more by their trust in certain advocates than by the technical merits of the various proposals. Those having good personal relations with board members were most likely to have their proposals for choice of vendors or kinds of equipment accepted, even when their adversaries within the firm had better cases on procedural or technical grounds.

Rational analysts often assume that technical criteria or the technical merit of system proposals will determine what computing systems will be selected by an organization. Software transfer, similarly, should be "natural," barring technical limitations or poor dissemination of transfer information, according to rational analysts. But the studies cited here point to the potent role that incentives, structural attributes of a computer-using organization, and organizational politics play in influencing whether computing is adopted, which technologies will be selected, how they will be developed, and whom they will serve.

2.3 Requirements Specification

Lately, within the computer science community, much attention has been directed at the capture and analysis of system specifications (Boehm, 1976). The current rational emphasis is to treat the capture of system specifications as a technical task which can be analyzed in a "structured" manner. But analysts utilizing other perspectives primarily attend to the interplay between users, specialists, and managers when developing system specifications.

The primary organizational concern for system specification activities, according to human relations analysts, is the closeness-of-fit to satisfying user needs. Oftentimes, satisfying users' needs is described as an important concern in system design (Hedberg and Mumford, 1975; Lucas, 1974). But a careful examination of this work reveals that emphasis on user involvement is really directed at specifying the functions and workings of an intended system:

specifying the system design.

According to human relations analysts, lack of involvement in specifying a system's design can lead users to be dissatisfied with their jobs (Lucas, 1978; Mumford and Banks, 1967; Mumford, 1972). Lucas (1974,1975,1978) argues that one recurrent reason why computing systems fail (i.e., fail to satisfy user desires) is that users are not directly involved in specifying the system's design. He also argues that smooth interaction between the users who specify and the specialists who develop the system is essential for achieving a satisfying and successful computing systems.

Mumford and Pettigrew (1976) claim that computing increases uncertainty among people in the organization and decreases their ability to cope with these uncertainties. Mumford stresses that if system designers (or specification analysts) attend to the "humanistic needs" of users, they will mitigate users' uncertainties and keep them satisfied when using a computing system. Attending to these needs should therefore facilitate organizational success with computing. But it is also quite clear that constraints on organizational resources and the distribution of computing expertise must be attended to in order to determine whether these needs can be met and the best ways to meet them.

In related work, Kling (1973,1977) examined the way in which emphasis on different system requirements translates into computing systems that are either "machine-centered" or "user-centered." Machine-centered systems reflect the focusing of system specifications toward efficient utilization of the computer. User- or person-centered systems reflect the focusing of system specifications toward minimizing computing hassles for users and improving users' job content. Kling (1977) has developed a conceptual framework which stresses both concern for the intended uses of the computing system, and sensitivity to local organizational politics and to the value-orientations of a system's designers, as a means for analyzing a computing system's focus.

Computing systems may be specified to reflect a variety of distinct interests. These include the perceived cost-savings of computerization, the problem-solving challenge for specialists to develop new systems and skills, reduced work demands and improved work effectiveness for instrumental users, and the perceived power payoffs achieved with computer-based organizational reforms. The specifications of a new computing system are shaped by the conflicts and agreements established between participants pursuing these different interests. The rational perspective suggests that conflicts over system design specification, with respect to interests served, are static, hence resolvable, rather than potentially problematic, fluid and ongoing. But interactionist and organizational politics analysts view these shifting conflicts as an intrinsic part of computing system specification dynamics. They suggest that system adopters may be as much concerned, if not more, with organizational gains, losses, and strategies for achieving favorable balances of resources and administrative influence as with specifying the technical intricacies of a computing system. The system developer thus may face a difficult dilemma in deciding which system "requirements" are to be further specified and met, and for whom.

2.4 Design

System design is often cast as an essential activity in the system life cycle. After all, if you can't build a computing system that is understandable and manageable, why should the system be expected to work well? Though such a concern may seem simple and obvious, practical design decisions depend in part on the size of the system and on the number of different people developing it. Many computing specialists are trained with systems that are relatively well-designed and small. For example, Kernighan and Mashey (1979) discount the utility of structured software design techniques for small programs (less than 1500 lines of code) when programming in a sophisticated computing environment. Similarly, most organizations probably have limited computing resources and technical expertise. In many large organizations, large-scale integrated software systems can require multimillion-dollar expenditures and many man-years of labor for development. In these settings, the efficacy of a system design can mean new savings and opportunities for the organization, or massive cost overruns.

Software design has emerged with a strong engineering influence, and a rational perspective which, at times, views software as a form of mathematics. So conceived, software design is thus a series of structured tasks executed by a hierarchically organized team of computing specialists (Mills, 1976,1977). The common problems of software systems which support a rationale for directing attention to system design activities include: the high cost of design errors (when the system doesn't function as specified); unfulfilled need for well-designed systems which are easily extended; and the absense of appeal to the intuitive correctness embodied in an engineering approach. Furthermore, engineering design strategies are preceived to provide benefits in later stages of a system's life cycle (e.g., a good system design results in easier-to-maintain systems (Boehm, 1976); system reliability is achieved through structured design and not through testing (Mills, 1976,1977)). However, the literature citing empirical demonstration of such claims is scarce. These techniques are usually promoted for their rational, but heuristic, value.

A finer examination of the engineering approach to system design reveals a casual concern for the actual users of a system. Part of this problem stems from the pervasive ambiguity of "who the users really are." That is, users can be either organizations, managers, specialists, instrumental users or clerical users. These people might each "use" a computing system in a very distinct way to realize different ends. Much of the software engineering literature tends to identify users as large organizations and their managers (Boehm, 1976; Brooks, 1975; Mills, 1976,1977). But little attention is directed at how users are to be involved and which users are to be involved in system design activities.

Judging by the literature on design activities, there is little understanding of the ease with which computing-naive users can participate in system design activities. Human relations analysts continually emphasize the importance of user participation is system design as the key to system implementation success (Lucas, 1974,1975,1978a; Mumford, 1967,1972,1975). Many computing users have little or no prior experience in system design activities; thus they cannot be expected to understand all of the technical exigencies that must be attended to during system development. Though these users may be more satisfied because of their participation, neither their interest nor their satisfaction guarantees that adequate system specifications will be captured. Similarly, participation does not guarantee that the implemented system will meet users needs or serve their interests; we may all be aware of how participation in organized activities can become merely passive or symbolic when outcome decisions have already been specified by others in control. Concerns such as these may be less at issue for technically skilled professionals but more at issue for casual or clerical users of computing systems.

Another approach to system design directs attention to the roles of systems designers and actual users as well as the intended administrative, social, and political uses of resultant systems in organizational settings. This approach represents a merger between the organizational politics and interactionist perspectives. From this spliced perspective, software design is viewed as an activity inseparably technical and social, taking place in settings where organizational politics, the negotiation of computing resource balances, staff sentiment and the like are all formative ingredients which shape computing system design, development and use.

Keen and Gerson (1977) outline the way in which the software system design process is embedded in the political order of an organizational setting. The political order of the organization shapes, constrains, or defines every phase of systems development and use. For example, conflicts over which system specifications are to be reflected in a system design may be resolved or disputed according to who has the greatest influence in specifying the design or use of the intended system rather than according to one's technical competency. These authors suggest strategic heuristics to help software designers to recognize and avoid political entanglements in the system design process.

The recent work of Kling and Scacchi builds upon many of the preceding findings of the social character of system design (Kling, 1978b; Kling and Scacchi, 1978,1979a,b). Computing system design is a recurrent activity in many social settings. Computing system designs are examined with respect to the present and prospective distributions of organizational influence and computing resources for managers, end-users, and system designers within an organization. Different system designs reflect the selected interests of those with substantial influence or resource control. Similarly, different designs when implemented can shift influence and resource patterns within an organization. But such shifts are constrained with respect to the proposed system, the "purposeful" activities of organizational actors, and the local computing package (Kling and Scacchi, 1979).[3] With respect to a computing package, the purposeful activities of organizational actors may include specialists trying to pursue rational system design techniques, managers attempting to extend their domain of control and influence, instrumental users trying to attend to job content and work deadlines, and computing promoters advocating the adoption of new or "better" computing technologies. The situated interplay of organizational actors pursuing their interests, the proposed system, and the local computing package outline the social character of computing system design (Kling and Scacchi, 1979a,b).

Situations like these often give rise to conflicting system design specifications. Depending on the influence of different users, the resulting system design will emerge from the resolution or stalemates of their system design conflicts. This, however, does not imply that such conflicts are monumental. Many will be resolved through deference to technical expertise, managerial authority, or administrative fiat. But system designers often can find themselves at the nexus of these interactions. Interactionist analysts thus point to the need to account for the meanings different actors give to system design, the social and technical resources at their disposal, the organizational setting and the organizational computing arrangements as inseparable features of the system design process.

2.5 Implementation

Much of the present discussion for system implementation comes from the Management Information System (MIS) arena. In this arena, implementation concerns the management of organizational change brought about through technological changes (Keen and Scott Morton, 1978). Presently, the introduction of new computing systems into an organizational setting is the major focus for system implementation research. The major issue pervading the literature is: will a project to implement an organizational information system succeed or fail. The research emphasis is often on assessing what features of an organization and its staff, and the computing system and its design, influence system success or failure *a priori*. Researchers often thus utilize a structural perspective in their analysis of system implementation.

Studies of system implementation addressed to computing specialists often rely on anecdote and "horror stories" (Brooks, 1975; Glass, 1977) rather than systematic empirical studies. These arm-chair analyses similarly deal with system implementation in terms of success or failure. For example, Brooks (1975, p. 47), commenting on the implementation of OS 360 notes: "It was very humbling to make a multimillion-dollar mistake, but it was also very memorable." Most of us are of course aware that OS 360[4] and its descendants are now found in many organizational settings. So why would it be considered a failure? An important distinction must be made in defining the success or failure of a system implementation. MIS analysts treat success or failure in terms of the achievement of *management* goals, whereas computing specialists are often attentive to *technical* criteria. Thus construed, a system implementation project could be considered by computing specialists to be a technical success, while failing to meet organizational or management goals; and similarly, a system implementation project might succeed in involving satisfied users and attaining management goals, yet be considered a technical failure. An interactionist analyst might therefore interpret this distinction to reflect the social meanings that different organizational actors can utilize in assessing the success or failure of a system implementation project.

Computer-based systems may be adopted by an organization, but still be difficult for staff to easily use. Frequently, casual observations are made about computer-based information systems that produce voluminous or unreadable reports. Addressing this difficulty of system use, human relations studies of MIS implementation repeatedly conclude that substantive involvement of computing users in system specification and design leads to systems which are better accepted. Unfortunately, "user involvement" has become a cliche, and both "user" and "involvement" are ambiguous in referent. The better studies differentiate kinds of "users": managers who occasionally see computer-based reports; staff analysts who generate reports; clerks who enter and retrieve individual transactions; and computing specialists who use tools or techniques to tune or maintain the performance of the system. Given this variety of users, we might infer that there exist divergent expectations about the potential for system implementation success or failure, and the nature of the expectations of a given user depends on the extent of his or her involvement. "Involvement" can range from being informed about design decisions made by technical staff, to actually creating design specifications. The research indicates that people who use computer-based systems will appreciate them to the extent that the systems help them meet their task demands and to the extent they feel some control over their system's behavior.

The empirical studies of MIS implementation generally focus on comparisons between the conventional wisdom of <u>how</u> to implement an information system and <u>what</u> is done in practice (Alter and Ginzberg, 1978; Keen and Scott Morton, 1978; Lucas, 1975,1978). Many of these studies indicate that practice does not usually follow academic wisdom. For computing specialists, for example, implementing an information system may be challenging, necessary, or a preferably avoidable set of tasks. We ask, why doesn't practice follow advice?

Rational analysts often slight system implementation activities. It is as if they define system implementation as just coding the system design. Thus construed, "implemented" systems should somehow naturally "slip" into existing organizational activities. But should the organization's implementation of a computing system be clogged up by unforeseen organizational problems, rational analysts will often site the need for "better management" to improve things. Structural MIS analysts often focus on those characteristics of the organization which must or will be changed to achieve the successful introduction of major technological changes (Alter and Ginzberg, 1978; Brooks, 1975). Human relations analysts, on the other hand, have documented the importance of user participation in system design to the achievement of successful information system implementation (Edstrom, 1977; Hedberg and Mumford, 1975; Lucas, 1974,1975,1978; Mumford and Banks, 1967). In addition, the organizational role and power of different users are posed as a complementary characterization to the human relations approach as the basis for the way user involvement is practiced and achieved (Mumford and Pettigrew, 1976). But analysts following an interactionist and organizational politics perspective would attend to how the implementation activity is operationally defined by different actors, the constraints on their involvement, and the placements of value on managerial versus technical objectives as characterizing features of the social process of system implementation.

2.6 Testing And Validation

Software system development practices often stress the importance of testing, validating, and verifying the correctness of a system (Boehm, 1976; Mills, 1976,1977). In theory, one would like to be able to generate automatically a set of test data sufficient to demonstrate the probable correctness of a robust class of programs and accompanying data sets. However, for programs of moderate complexity, the set of test data to exercise all paths through a program is infeasibly large. Some promising research is proceeding on various schemes to automate tests for special program conditions. However, basic questions have been raised as to whether people will trust an automated program tester or verifier, because of what it takes in practice to convince someone of the efficacy of automated "proofs" (DeMillo, Lipton, and Perlis, 1979).

DeMillo and associates argue that the rationalist appeal for viewing software as a part of mathematics (Mills, 1976, 1977)--thus facilitating the application of "formal" theorem-proving techniques--is confused and misconceived. DeMillo and company sympathetically agree with the appeal and robustness of the mathematical perspective, but for non-rationalist reasons. They argue that mathematics is, rather than a formal process, an ongoing, interactional social process.

In contrast to the research on new systems or techniques for program testing, the current state of practice does not rely upon much automation at all. Computing systems are "proved correct," validated, or thoroughly tested, for example, either by using the system in production or by selectively demonstrating the system's capabilities with data "manually" selected by a specialist or knowledgeable user.

An adjacent matter to that of software testing and verification concerns the accuracy, reliability and auditability of results processed with automated systems. For example, though a program (or parts of it) may be shown to behave "correctly" for a given test data set, ensuring the accuracy of data processed by many individual users across many other programs and organizational units leads to practical difficulties. These difficulties become most apparent when one attempts to reconstruct what processing occured and when. These are often practical concerns of organizational computing users who are accountable to clients and governmental agencies as well as other computing specialists and instrumental users. Such topics are often of interest to analysts of organizational politics. But these topics have not yet garnered much attentive research within the computer science community.

2.7 Documentation

System documentation ususally is intended to function as a source of information for what a given computing system is good for and how to use it. Rational analysts argue that the preparation of documentation should not be separated from system development. Furthermore, they argue that good interactive text processing facilities permit quick and convenient production of many kinds of documentation that might otherwise be unobtainable, impractical, or very expensive (Mashey and Smith, 1976).

When a new computing system is introduced, its users benefit from both tutorial manuals and reference manuals. Documentation used to supplement user training activities usually provides demonstrations of exemplar processing tasks. System specialists, who must maintain a system, can use good design and program implementation documents. Such documents may in fact include routine procedures for (limited) data and program testing. However, adequate and up-to-date software documentation of these kind is continually a weak feature of most software systems (Kling and Scacchi, 1979a,b). Our observation is that it is rare for a computing system to have up-to-date, high quality documents of all these kinds.

Software system manuals can often be measured in inches, but their currency and adequacy vary. Palme (1978) has reported that though documentation may abound for individual programs, documents which describe how multiple programs can interact under user direction are unavailable or effectively unretrievable from the documentation morass. Maynard (1979) has indicated that some users want manuals that are less wordy while others want manuals with more explanations. Belady (1978) and Brooks (1975) have indicated that, as an organizational task, the sheer production of system documentation can consume significant amounts of organizational materials and services. In addition, the comprehensibility of computing systems with substantial documentation of varying quality has been raised as a practical organizational, technical, and person-centered concern (Belady, 1978; Brooks, 1975; Kling and Scacchi, 1979a,b). According to interactionist analysts, providing practical and up-to-date documentation demands time, attention, skills in clear and concise writing, an inclination to help users, and an organizational commitment to encourage and support documentation activities (Kling and Scacchi, 1979a,b).

In practice, documentation work is subject to many of the contingent demands that can pervade an organizational computing setting. Interactionist analysts observe that the people who produce or use system documentation manage or shirk the contingent demands for documentation upkeep and adequacy. On the other hand, rational analysts suggest that such dilemmas don't arise in resource-rich computing environments where suitable on-line documentation and storage are available (Kernighan and Mashey, 1979). But it is questionable whether most organizational computing environments are resource-rich. Though the rational analyst might then argue for computing resource redistribution, organizational politics, interactionist and perhaps structural analysts would counter that such redistributions are not achieved without attendant organizational conflicts.

2.8 Maintenance, Modification And Conversion

Current system life cycle costs reflect the high and growing cost of system maintenance (Belady, 1978; Boehm, 1973; Lientz, et al, 1978). The organizational resources spent on system maintenance and modification are often much larger than expected. Furthermore, maintenance costs are difficult to predict or determine after the fact because these costs can be distributed across or hidden within an organization's budget. The traditional concern, as primarily addressed by rational analysts, is on the high cost of software maintenance rather than the nature of the activities that constitute it.

The available figures for system maintenance do not distinguish the costs of system alteration in terms of time, skills, inclination, forsaken opportunities, attention, and other organizational resources or their distribution across an organization (Boehm, 1976; Wolverton, 1974). Similarly, the figures do not account for how these social resources are consumed by users and managers interacting with specialists trying to implement the requested enhancements in an already complex computing system. For example, in order to convert a related collection of data files--a data base--over to an integrated data base management system, we might assume the specialists doing the conversion have the necessary knowledge of the contents (and structure) of the data base (Oliver, 1978). But if the data base has been in the organization longer than the conversion specialists, then these specialists must rely on information provided by others to implement and test the conversion. This information may be in the form of either system documentation or verbal information from others. Neither of these sources is necessarily reliable, up-to-date, or sufficiently detailed. System documentation, as we have already observed, is usually described as being of poor quality and out-of-date, while deriving information from others can sometimes be problematic because of (a) differences in perspectives toward the uses of a data base, (b) the specialists' concern for detailed structural information versus the users' concern with organizational procedure, and because (c) the data base may be sufficiently large and complex to be incomprehensible by any single person. Interactionist analysts note that these organizational dynamics and dilemmas may contribute significantly to the high cost (and organizational distribution) of system maintenance (Kling and Scacchi, 1979b).

In order to better control rising maintenance costs, some analysts recommend rearranging system development and maintenance work of programmers. Also, they suggest managing programmers with "proper" supervisory control. In other words, they recommend altering or increasing staff resources and management attention. These suggestions conform to a structural analysis.

Rational analysts also suggest the importance of "better management" and improved "programmer motivation" as partial solutions to the system maintenance problem (Cave, 1978; DeRoze and Nyman, 1978). But these are often vague and expensive remedies to recurrent computing work problems. And such remedies may not be readily implemented because of contending demands (and battles) for finite organizational resources and staff attention, according to organizational politics analysts. The contingent needs of organizational groups to get software systems maintained and system maintainers more committed to meeting user requests may be better understood by attending to work group interactions and resource contraints, according to interactionist analysts.

Currently, participants within the computing community emphasize software system development over system maintenance (Kling and Scacchi, 1978,1979a,b; Learmonth and Merten, 1978). Many development activities are based on the attractive assumption that good system development practices can significantly reduce system maintenance costs. Rational analysts, for example, claim that demands for software maintenance activities in resource rich, cooperative work settings are minimal due to an abundance of software development tools and an integrated programming environment (Kernighan and Mashey, 1979). But demands for system alteration can come from organizational actors far removed from system operation.

Computing systems are often treated as extensible entities. As such, older, embedded systems may be revitalized and extended by altering existing subsystems or adding new ones. In this manner, systems are developed <u>layer upon layer</u>: each layer adding an increasingly complex web of interacting system modules (Scacchi, 1980). But many systems become distended, procrustean, and fragile after numerous repairs, adaptations, and extensions. Understanding the workings and intricacies of such embedded systems becomes more difficult, thereby complicating system maintenance tasks.

Though software systems may have no physical moving parts, systems apparently "wear out." Wearing out is another way of saying that a system is too complicated or costly to use and maintain. Depending on the importance of a worn out system, coalitional or individual efforts may be initiated to convert the old system to a new one. The availability of new technological developments may further encourage some organizational actors to initiate system (and data) conversions or to adopt newer systems (Oliver, 1978). Similarly, the managerial desire to reorganize a perceived fragmentation of staff interactions and computing work can be another force acting to bring about the demise of an old system and the initiation of a new system. These conversion activities help recreate a computing system "life cycle." Rational and structural analysts tend to slight or ignore the practice of converting worn out systems into improved, often more complex systems. Their analysis, which may promote or assess the efficacy of computing system adoption, often treats system initiation as an activity discrete from (if not independent of) the conversion of older systems. Interactionist analysts, on the other hand, assert that the organizational and technological dynamics that surround computing system maintenance, conversion, and initiation characterize, in particular, the continuous production of computing system life cycles (Kling and Gerson, 1977; Scacchi, 1980).

2.9 Evaluation

With the ever-growing complexity of computing systems, problems related to system performance evaluation have received variable attention from researchers and practitioners alike. Interest in computing system evaluation is usually split between specialist and management concerns. For computing specialists, systems are evaluated according to some set of performance measures. These are generally diagnostic, but machine-centered measures. For example, hardware or software monitors may be used to measure processor utilization and operating system scheduling policies and analysis of the derived measures can be used to support decisions regarding hardware enhancements or software tuning. Use of system performance measurements can thus "create" certain demands for system maintenance. Management concerns, on the other hand, address evaluation of the effectiveness of a computing system in fulfilling "hard" or "soft" organizational goals (Keen and Scott Morton, 1978). These evaluations are typically post facto rather than diagnostic (Keen, 1975). For example, UMIS in Riverville can be evaluated in terms of how well it improved caseworker productivity (i.e., number of cases processed per unit of time/person), or how well it integrated welfare services. But the assessment of how UMIS enhanced administrative credibility in Riverville is an evaluation not likely to be conducted by management, at least publicly.

People who work with computing systems are as much part of an organization's information processing system as is the computer system itself. Computer performance measures which "objectively" monitor machine-centered system activity discount or ignore the import of instrumental users' concerns. Rational analysts often assume that maximum or optimal system utilization is the important user need. But it is often computing specialists or computing facility management who decide whether a system utilization pattern is suitable or constrained. It is these people who decide what measurements to make and how to interpret them. And though we might believe that such decisions are always based on competent, technical reasoning, there is no empirical basis which support the accuracy of such a belief. In fact, we might as easily assume just the opposite: decisions concerning effective computing resource utilization are based on ad hoc or impressionistic assumptions of computing specialists or managers. Interactionist and organizational politics analysts might argue that specialists' or managers' use of system performance measurements, selective even though well-intended, can serve to legitimate attempts on their part to establish or reconfigure computing resource utilization patterns according to their desires. The accuracy of either of these system performance management perspectives awaits further empirical study.

One management approach to system evaluation is to quantify and assess the costs and benefits which accrue from an organizational computing system. Computing systems are often initiated, designed and implemented with the outlook that certain intangible or qualitative benefits will occur. Rational analysts often claim that improved managerial control, "better" and more timely information, improved decision support and the like are the desired benefits (Bernard, et al, 1978; Champine, 1978; Simon, 1973; Walston and Felix, 1977). But a thorough accounting of computing costs should note cost displacements, and should reveal the presence or absence of real cost-savings. Cost displacements are "hidden" costs borne, for example, by users when hassling with poor documentation or when trying to figure out why a system doesn't behave as expected. In practice, such accountings may not be done regularly. But it is often assumed that the outcomes of such accountings will reveal real cost savings for computing use over alternative techniques. Again, we must await for empirical assessments.

In general, then, computing work is assumed to be cheaper than manual work for comparable tasks, but, in practice, this assumption is not systematically checked. Furthermore, it is not clear that if such an accounting were performed which revealed the lesser expense of manual techniques, users would or could readily drop computing and move to the "rational" cost-efficient method.

2.10 CSRO Case: A System Life Cycle

In the early 1970's, a computer science research organization, "CSRO," acquired a large-scale time-sharing computer system to support their ongoing activities in various aspects of computer science research. We studied this computing setting during 1977-78, well after the computer system had been installed and usage patterns rooted (Scacchi, 1980). In this section, we will examine CSRO's text processing system.

The computer system at CSRO is linked to similar organizations and computing facilities through a national computer network. The computer system utilizes virtual memory in addition to 2+ megabytes of main memory and 100+ megabytes of disk storage. The system has demonstrated its capability to support more than 50 interactive users during peak use times. Some users of this computer system have developed very large interactive application systems written in a popular dialect of the programming language LISP (Sandewall, 1978). In short, the CSRO computing environment was quite sophisticated, "state of the art."

The automated text procesing system at CSRO is regularly used by project managers, computer science researchers, system programmers and secretarial staff. Project managers, usually senior researchers, use the complete text system infrequently except when work deadlines necessitate its use. Managers take advantage of their organizational position in allocating secretarial or junior staff to certain text processing tasks. These tasks usually amount to executing the computational tasks necessary to obtain a finished version of the source text document. Such a computing work allocation scheme often leads the secretarial staff to use the system on a daily basis. The systems programmers and other CSRO researchers use the text processing system regularly, as they need it. These junior researchers and staff often must individually tend to all

Most users rely on the help of other experienced users in order to manage or minimize NEAT hassles. One user commented: "there a lot of bugs in it. But people who use it regularly have charted out the bugs which is sort of nice because (then) you know where they are." Another user stated: "Most people around here treat it (NEAT) as a black box. You usually go to one of the people around here who knows it to avoid trouble." It puzzled us why the presence of system bugs should pose recurrent difficulties in system use. Why weren't these bugs readily fixed? According to one systems programmer, "actually, it's not maintained at all." What resulted was that many different sets of NEAT macros were implemented by capable users to get around the bugs they encountered. However, many of these NEAT macro routines were incompatible or redundant. This gave rise to new system maintenance demands--upkeep of NEAT bug-avoidance macro libraries.

The text processing system came to our attention in part because of its regular use by most CSRO users. The "system" was composed of different software systems: "line-oriented" and "screen-oriented" text editors (four now in use); a spelling checker; text formatters (two in use, one in development); and supporting systems software, which handle source file "movements" and output listings. In addition, on-line CRT terminals and high quality printers constitute part of this system.

The users at CSRO include graduate student research assistants, research associates, academic faculty, project managers, and computer programmers--all computing professionals. Many users characterized the need for a text processing system as "essential," given their desire to produce "professional" documents or publications, in light of the limited secretarial services at CSRO. Some users were quite concerned with their ability to control most or all aspects of document production, and some of them indicated that their use of the text processing system reduced or eliminated the need

The design history of these various text processing tools was somewhat more difficult to discover. None of these software systems was designed or developed by anyone within CSRO; thus our history of the text processing was fragmentary. Some users knew who designed and built these software components. Some also readily pointed out what they considered to be system design flaws. System bugs, missing features, irrelevant system features, awkward stylistic conventions and poor system performance characteristics were pointed out by users. Nonetheless, they used the system.

The implementation of the text processing system follows a trajectory similar to that of system design, that is, sketchy and fragmentary. Some of the system flaws may have resulted from the method of implementation by the system builders. There was no overall system design to integrate the different system components.

Many users had implemented their own version of one of the text formatting programs. Users had accessed the original "public" version of the program and altered it to meet their own needs. Users said they followed this strategy because they could identify and repair (and hence understand) program bugs and the program itself. Furthermore, a user could to some extent add, subtract, or revise program features given his or her own version of the program. But different users indicated that no one at CSRO really knew everything about how the original program worked.

Nearly all users reported that certain software components, particularly the text formatter they preffered, were rife with bugs. Most bugs were found through testing with real data. That is, the system wasn't sufficiently tested by specialists or users _a priori_. (In practice, users attempt an action they believe to be correct, as indicated by someone else or by the system documentation, and find that some unexpected system behavior results.) The documentation for NEAT was roughly five years old and not updated to reflect maintenance alterations. This led to some problems for users who acquired an intermediate version of the NEAT program that had been altered without the documentation also being changed to reflect this. According to the users, other software components did not have equally bad documentation. But an opinion expressed repeatedly about all software documentation was that it could always be better.

As we have noted, some analysts have suggested that one set of recurrent problems in maintaining computing systems is attributable to a primary emphasis on issues of system development. The traditional practice has been to focus on system development activities as opposed to system maintenance. The apparent underlying rationale for this position follows the belief that if a computing system is developed in an orderly manner, then naturally the resulting system will be more easily maintained. But this rationale disregards the problem of maintaining systems that have been previously developed in a less than orderly manner. This latter position seems to be the more appropriate characterization of the development of the text processing system at CSRO. Given its development path, what issues in system maintenance have surfaced?

We saw in the case of the NEAT subsystem that system bugs and idiosyncracies are not necessarily repaired. No one was assigned to maintain the NEAT system. One user strategy to avoid, circumvent, or undo adverse system effects was to develop manageable NEAT macro routines. Another strategy was simply to chart out the location of known system deficiencies and to structure the content of text processing computing tasks to avoid problems. Bug charting activities can entail relying on the experiences of other users who have passed through similarly troubled waters. The point drawn from both of these strategies is this: the fact that design or implementation errors are known to exist in a given computing system does not necessarily mean that the system will be repaired; users will alter their computing work styles to get around the limitations. In addition, since many users develop their own sets of mistake-avoiding software routines, these routines must also be maintained. These routines are likely to conform to the particular needs of some, but not all, users. For the text processing system at CSRO, system maintenance activities are distributed across the user community.

Many computing analysts report the increasing cost of software system maintenance. At CSRO, the text processing system is regularly used by many users. Maintenance on this system, however, does vary. Earlier, we described how the text processing system developed from two text editors and one text formatter to encompass four text editors, a spelling checker, and two text formatters. There are now more individual subsystems to maintain and unless members of the CSRO user community are somehow persuaded or restricted to use only some subset of these, they must all be maintained. The necessary maintenance may be distributed throughout CSRO and not confined to the systems programmers--those normally assigned to system maintenance--but it is unclear whether the cost of maintaining the system will increase. That is, while there are more software components to maintain, a major portion of maintenance activity is subsumed by users distributed throughout the organization. In addition, their maintenance activities may not get directly accounted when system operation and maintenance costs are assessed. Such a system maintenance arrangement may be common practice but it can be deceptive in that it can increase "hidden" costs. And under this maintenance arrangement, personnel turnover can act to further exacerbate the costs of maintaining existing systems.

All organizations experience personnel turnover. And as different people pass into and out of the organization, so do experiences, understandings, and macro libraries for different computing systems. In a case where understanding the workings (and bugs) of a given computing system is spread across the user population, we might expect that new organizational members (or infrequent computing system users) will have to spend some time acquiring a working knowledge of a system. Again, such arrangements may be common or expected, especially in a sophisticated computing environment. But what does this say about the cost of system maintainence? System maintenance is usually accounted for according to expended resources: labor, computer time, organizational overhead, etc. We might expect that greater personnel turnover will drive up the cost of system maintenance because more people have to take time (labor and possibly computer time) to learn the system. CSRO is set up to regularly "turnover" skilled graduate students and visiting research associates. Furthermore, when organizational knowledge of the workings of a given computing system is spread across many computing users rather than confined to system programmers and good documentaion, the ease and frequency of interactions between instrumental users and others will influence the content of system maintenance and hence, maintenance costs.

Finally, to what extent do system maintainers avidly pursue or desire system maintenance work? In the words of one system programmer at CSRO: "computer programming is a lot of fun, like solving puzzles. But there is a lot of shit work around that you have to get involved with: documentation, inheriting somebody else's program that is rife with bugs. These are not fun." Thus we were not surprised to find users distinguishing maintenance activities from "real" computing--that is, system use or development. The cost impacts of these conditions are difficult to quantify, but it is not clear that these arrangements can be ignored in determining what affects system maintenance costs.

Concerns for system maintenance naturally led us to address matters of system evaluation. How well does the CSRO text processing system perform? We solicited evaluations from users on the practical utility of CSRO's text processing system.

The general user sentiment about automated text processing was quite positive. Text processing was said to be one of the best benefits of interactive computing. In the words of CSRO's computing facility manager: "it's a very powerful tool, especially for preparing large documents." Though there was much apparent support for text processing, many users cited their reservations. Three CSRO users, each working in different research groups, commented respectively:

"There is a lure to text processing. It's very seductive to keep hacking at a paper to make it look nicer since you have the ability. I may get more work done if I used a secretary, but not as nice looking or reading as clear."

"The process of making text pretty is enormously time-consuming. (but) it's getting to the point where a secretary, rather than a programmer can do it."

"As things have got fancier, the first (system) introduction seems powerful. But as you advance (to other systems), it seems hard to step down or back to the others."

From these remarks it is unclear what the real organizational or performance benefits (e.g., improved appearance of research documents) of automated text processing are. There are some conflicts between the generally positive sentiment and these individual assessments, which appeal to different system values.

We may each draw conclusions about the efficacy of this computing system, its users, or even CSRO. But let us turn to review this case data with respect to the six perspectives on computing in organizations, outlined in section one, to gain further insight into the nature of an organization's computing system.

2.11 Six Perspectives Of Text Processing At CSRO

In initially reflecting on the data on CSRO's text processing system, we each can draw a variety of conclusions. We may, however, discount the viability of a computer science research organization as our object of examination because it is a research organization and hence atypical. But CSRO, like any organization with computing systems, can be examined for those of computing activities that are involved in the production of organizational products. We also might have to hedge our analysis by the fact that CSRO is a computer science organization and thus atypical. Our concern here would be that organizations consisting primarily computer scientists are somehow different. But that same challenge must apply to any organization populated by some particular group of professionals.

Instead, we propose to review the life cycle of the text processing system at CSRO, using the six perspectives on computing in organizations. These perspectives might not provide suitable insight into each phase of the system life cycle, but each perspective might lead to deductions or insights which can be contrasted with those drawn from the others. Thus we should be able to enlarge our general understanding of the dynamics of text processing at CSRO, of computing in a professional and sophisticated setting, and of the social character of a computing system life cycle.

Rather than go through the issues raised by each perspective, we provide a table of representative questions which the reader can choose to answer given our previous discussion of each perspective and the CSRO data just presented. These questions exemplify the concerns of analysts adhering to different perspectives in understanding the life cycle of CSRO's text processing system.

Place Table 3 about here

2.12 A Retrospective Of The System Life Cycle

We believe the CSRO case supports our earlier claim that computing systems evolve by way of a process of layering throughout their life cycle. Computing system components and their interconnections are reassembled, reconfigured, and enhanced by instrumental users as well as by computing specialists. People _tinker_ with a computing system to make it do what they want it to do (cf. Jacob, 1977; Scacchi, 1980). At CSRO, some users took it upon themselves to acquire their own version of NEAT, which they would enhance to fit their needs. The evolution of a computing system may thus be a process of tinkering and layering. That is, users' tinkerings will accumulate in many shapes and forms--some by design, some by hasty effort, and others just obscure or bizarre--which coalesce and emerge as new layers of system.

When many system layers accumulate and their interactions tinkered beyond clarity, these and related aspects of a computing package will be "defined" to be unmanageable. (At CSRO, the general quality of documentation for NEAT degrades as multiple-version enhancements are made and as haphazardly documented macro libraries build up, unless systematic efforts are undertaken to restore it.) As this occurs in an organization, efforts will emerge to stop using the system, to convert it, or to replace it with some newer arrangement. These efforts denote the passage of a computing system from the end of it's life cycle to the beginning of another. It also denotes a passage in the evolution of an organization's computing package. Thus, we expect that in the near future, NEAT will be seen to be decreasingly useful throughout CSRO, especially as new, more encompassing components appear which subsume many of NEAT's enhanced capabilities.

It appears that instrumental users may have to take on a significant portion of system maintenance tasks in order stay abreast of their growing and contracting computing environment. However, the skill with which these people tinker with the layers of a system will shape the ease or hassle of computer use, and probably a major share of system maintenance costs. If instrumental users do not take up maintenance tasks, then these users must renegotiate or reshape the work content to conform to the contours of the computing environment. In this later arrangement, users bear the costs of poorly fitting computing systems. Will advances in system life cycle engineering improve this situation? We believe that depends on what perspective(s) computing system developers will utilize when conceiving the development of new, or the conversion of old, systems.

Table 3

Six Perspectives on the CSRO data

Rational -

1. in what manner is the text system a labor-saving device?
2. how does the text system augment current secretarial service limitations?
3. how is the automated text processing better than conventional techniques in the ways it is "naturally" used?
4. how would automated documentation tools aid in resolving current documentation problems?
5. according to what criteria can system performance be measured?
6. are text productivity impacts positive for the text system? If not, is the system efficient?

Structural -

1. are text production activities now more centralized? What difference does/would this make?
2. can users more readily manipulate and produce complex documents with more speed and ease?
3. can more users utilize the existing text system as compared to a manual system?
4. what attributes of the organization and the existing computing facilities "encourage" the use of the system?

Human Relations -

1. are users satisfied with their use and control of the system?
2. are users troubled about not being involved in specifying the design of the text processing system?
3. does the use of the text system provide some motivation for its continuing use?
4. does the text system help users cope with certain uncertainties (e.g., ease or timeliness of producing documents) in their work?

Interactionist -

1. what import do CSRO text users assign to the text system?

2. do text users give contradictory definitions of text production activities on the one hand in contrast to the efficacy of the text processing system on the other?

3. how do users view the text processing system as a social object and not just a technical artifact?

4. what interpersonal negotiations take place which shape the different phases of system development?

5. what social and technical resources are bartered in these negotiations?

6. why do text users perfer to use the text system in spite of their feeling that manual alternatives (secretaries) may be a more efficient way?

7. why are text system users willing to reshape the style or content of their work to conform to the "contours" (including system limitations) of the system?

8. to what extent do users sense greater negotiating ability or control over text processing because of the availability of multiple system configurations?

Organizational Politics -

1. what resources do CSRO text system users command and how do they use these to increase their power?

2. do CSRO's managers make use of the text system to enhance their control over CSRO operations or products?

3. to what extent are managers able to extend their control others at CSRO because of the text system?

4. what are the relations of organizational power and dependency that surface in the use of the text system?

5. what highlights the text system's utility as an instrument of bureaucratic politics for dealings with outside organizations (e.g., preparing grant proposals and project summaries as production activities conducted or supervised by CSRO managers)?

Class Politics -

1. how does the use of the text system at CSRO deskill the quality of work, knowledge, or control over computing or other work activities for the users?

We have tried to show how analysts following different perspectives define and assess problems in the system life cycle. We believe the interactionist perspective captures the richest description, and provides the most "active" analysis, of the nature of these activities. The analysis within this perspective suggest that people will shape their role in the different activities of a system life cycle in order to complete their computing tasks with the least hassle unless what they value is threatened. Other analysts point out that many "technical" activities in the system life cycle are influenced by workplace incentives, who controls key decisions, whose interests are served, and how these are mixed and meshed -- organizational politics. Human relations analysts tell us that success or failure in fitting a computing system into its organizational environment depends on involving users in specification, design, and implementation of a system. Structural analysts note that attributes of the structure of a computing system, the organization in which it is embedded, and the external environment of the organization also play a part in determining the fit of a system. The rational position, unfortunately, attempts to limit our attention to just the technical nature of activities in the system life cycle. While such technical matters often pose challenging problems of their own, we believe that constant attention to a technological or "systems" point of view denatures our understandings about what different people can and will do to make a computing system habitable and useful. Finally, we note in passing that the class politics presently provides no contribution to understanding the dynamics of a system life cycle.

We will now move on to a section which examines a subject intimately entwined with a system's life cycle: computer system use. In this next section, we will present a social analysis of computer system use in complex organizations.

Banks, 1967). The work of clerical users is generally managed in such a way that they must use a computing system as provided. Thus patterns of computing use are characterized by differential discretion in computing use on the part of organizational actors.

Managers are designated with organizational decision and policy-making authority. When it comes to computing, few managers actually use a computing system on-line. Rather, they usually delegate information retrieval tasks to subordinate professional or clerical staff. If a given manager is particularly influential, either because of organizational responsibilities or power base, then the manager can often get his computing tasks assigned high priority. Similarly, tenured organizational staff may be able to utilize this kind of influence.

While we all may recognize managerial prerogative, managers can utilize their organizational authority to assure their subordinates easier access to computing. Managers in organizations can and do influence other users' patterns of computing use. Though managers use computing primarily for routine needs, they may occassionally need to exercise their prerogatives for computing use. In small organizations with few sub-units, this poses few difficulties. However, in a large organization with many sub-units and special projects, unexpected managerial prerogatives for computing use can interrupt some computing tasks for different instrumental users in that organization.

Structural analysts might point to how a manager's "span of control" determines staff access to management-oriented computing use. Human relations analysts assert that people value what they control (Lawler and Rhode, 1976). Thus, managers should value their ability to control staff access to management-oriented computing uses. But instrumental users, clerks and specialists will also value computing when they have

3.0 COMPUTER SYSTEM USE IN COMPLEX ORGANIZATIONS

Empirical and theoretical studies of computer use range in focus from promotional speculations to analysis of the recurrent dilemmas of computer use in organizational settings (Kling and Scacchi, 1979b). In this section, we will focus our discussion to four facets of computer use: discretion in computing system use; the distribution of knowledge for computing system use; commitment to computing system use; and mistakes in computing use. These will be examined with an eye to the six perspectives on computing in organizations.

3.1 Discretion In Computing System Use

Organizations provide an arena for interaction among computing users of all kinds. Managers in organizations use computing selectively. They most often use computer-generated reports for routine organizational decision-making, rather than directly accessing terminals or programming their own applications (Dutton and Kraemer, 1978). Instrumental users such as accountants, urban planners, insurance actuaries, etc., use computing to routinely analyze data and prepare reports for administrative purposes. Occasionally, they conduct exploratory data analysis (Kling, 1978e). In addition, managers and instrumental users often have the <u>option</u> of using computing; computer use may not be desirable for all tasks given individual or organizational interests and limitations.

Computing specialists are generally eager to use and further develop a given computing system. The amount of attention specialists give to a system or a particular application depends upon the the demands for scheduled service delivery and personal or professional interests. Clerical users, on the otner hand, are constrained in their use of a computing system to data entry and/or retrieval (Mumford and

Banks, 1967). The work of clerical users is generally managed in such a way that they must use a computing system as provided. Thus patterns of computing use are characterized by differential discretion in computing use on the part of organizational actors.

Managers are designated with organizational decision and policy-making authority. When it comes to computing, few managers actually use a computing system on-line. Rather, they usually delegate information retrieval tasks to subordinate professional or clerical staff. If a given manager is particularly influential, either because of organizational responsibilities or power base, then the manager can often get his computing tasks assigned high priority. Similarly, tenured organizational staff may be able to utilize this kind of influence.

While we all may recognize managerial prerogative, managers can utilize their organizational authority to assure their subordinates easier access to computing. Managers in organizations can and do influence other users' patterns of computing use. Though managers use computing primarily for routine needs, they may occassionally need to exercise their prerogatives for computing use. In small organizations with few sub-units, this poses few difficulties. However, in a large organization with many sub-units and special projects, unexpected managerial prerogatives for computing use can interrupt some computing tasks for different instrumental users in that organization.

Structural analysts might point to how a manager's "span of control" determines staff access to management-oriented computing use. Human relations analysts assert that people value what they control (Lawler and Rhode, 1976). Thus, managers should value their ability to control staff access to management-oriented computing uses. But instrumental users, clerks and specialists will also value computing when they have greater control over its use and orientation. Organizational politics analysts are quick to point to the pending conflicts between managers and others over control of staff access to computing use and computing job schedules. Additionally, these analysts could point to the conflicts stemming from the conflicting requirements on a single system made by managers, users and specialists. Interactionist analysts, on the other hand, seek to understand and distinguish (a) when managers utilize their computing prerogatives, (b) how the distribution of managerial authority influences their utilization, (c) which conflicts are rooted in fixed versus negotiable commitments to schedules and system (user) orientation, and (d) how and when others are able to control computing use given managerial controls.

Users continually request computing system alterations even for routine computing applications (Lientz, et al, 1978). These alterations are requested for the purpose of meeting changing conditions in the organization's environment, and they are implemented by specialists who follow some scheme for prioritizing changes and for scheduling commitments of organizational resources. The priorities and schedules are sometimes established or arbitrated by an intervening administrative group (e.g., a "change control board") of representative managers, users and specialists. In order for users to get a system change request serviced, they may have to meet and coordinate with their managers, other users, the change control board and the specialists who will implement the change. Thus, as interactionist and organizational politics analysts show, the negotiations among system users, change controllers and specialists over system enhancements mutually affect the patterns of computing use and the contingencies of computing work.

3.2 Organizational Distribution Of Computing Knowledge

Our preceding observations on the discretionary use of computing point to a differential distribution of computing knowledge within organizations. How do different distributions of computing expertise facilitate or obfuscate easy, hassle-free computer use?

Rational analysts suggest that because computing systems are "rational," users can rely on personal cognitive abilities, the computing system, and system documentation to facilitate computer use (Kernighan and Mashey, 1979). In this way, the distribution of computing knowledge is localized and "efficient." Alternatively, interactionist analysts suggest that users rely primarily on information gained through dealings with other users, system specialists, or personal experience (Kling and Scacchi, 1979b). From this perspective, computing expertise is organizationally decentralized but "effective." Interactionist analysts argue that when a computing system and it's documentation lack coherence and comprehensibility, users will turn to others for help to complete computing tasks. Many computer scientists point to the dilemmas that fragmentary and dispersed computing knowledge creates for sophisticated computing use (Brooks, 1975; Palme, 1978; Kling and Scacchi, 1978,1979a,b).

Two related social features of organizations affect the distribution of computing knowledge. These are ease of access to computing expertise and the level of commitment to computing. Having easy access to computing expertise is a concern for many instrumental computer users. The level of commitment to computing helps distinguish computing specialists from instrumental computer users.

3.2.1 Access To Computing Expertise -

Instrumental users often interact with other users and specialists. These interactions sometimes arise from the need to resolve inadequacies in the system documentation. At other times, users need assistance to understand how to process "special cases" or "exceptions" given the range of system functions normally provided. In rich computing environments with many competent and cooperative computing users, computing expertise is often readily available. In most computing settings where computer use is scheduled or regulated, access to computing specialists is mediated through a chain of liaisons or supervisors (Kling and Scacchi, 1979b). Some specialists prefer this arrangement because it "buffers" contact with end users. Users, on the other hand, sometimes covertly bypass liaisons in order to get access to particular specialists familiar with their computing needs (Scacchi, 1980). While access to system documentation is generally provided, access to specialists knowledgeable in the workings of a given system is restricted but possible.

Structural, human relations, and organizational politics analysts suggest that the distribution of computing knowledge together with bureaucratic mechanisms (such as chains of liaisons between specialists and users) can either smooth or muddle easy access to computing (Danziger, 1977,1979; Mumford and Pettigrew, 1976; Pettigrew, 1972). These analysts point to the nature of particular organizational computing arrangements as a major influence affecting the ease or bureaucratic hassle of computing use. However, rational analysts seem to treat the distribution and access to computing knowledge as non-issues because of the tacit assumption that all users have equally strong interest in computing.

3.2.2 Commitment To Computing System Use -

It seems to be the case that specialists have an extensive commitment to computing, while instrumental users often have a much weaker commitment. This is indicated by the fact that instrumental users have been known to rely on others, including (sometimes to their chagrin) specialists, to help complete computational tasks. Specialists, on the other hand, seem to identify professional expertise with their ability to read raw code in order to "really" find out how a system works. In fact, system specialists are often surprised when users seek their help to acquire information that can somehow be found in the system documentation. Although some instrumental users may occasionally look at system code--often the only up-to-date form of "documentation"--it can be supposed that, in general, they do not share the detailed understanding or interest in software system intricacies that are earmarks of commitment. If all system users were equally committed to computing, then all users could be expected to read system manuals to learn the system.

The rationalist ideal is that a user thoroughly understands a system before using it. But while specialists can usually be expected to be attentive to the intricacies and details of a system's workings, instrumental users frequently have other interests and organizational commitments beyond computing system use. For this reason, instrumental users seem to be willing to learn those functions and commands necessary for specific computing uses as long as they can do so without expending more than the smallest effort.

For interactionist analysts, a user's commitment to computing involves both social and technical computing resources. The social resources include time, information, skill, budget, inclination and opportunity. The technical computing resources include memory space, disk storage, software utilities, processor time, computer terminals, etc.

These analysts note that negotiations between users and specialists over computer use and computing work focus on altering one's computing resource commitments to accommodate the needs of others. Users' and specialists' respective resource commitments to computer use are but one set of demands or investments that each attends to in completing computing tasks. The computing resources that are users' and specialists' objects of negotiation or commitment shape, in part, organizational computing arrangements and computer use patterns.

3.3 Mistakes In Computing System Use

Computing system use often results in the regular occurrence of "mistakes" at work (cf. Riemer, 1976). Computing use mistakes fall into four categories: miscalculations (such as enhancing the wrong code module, deleting the wrong disk files, changing the wrong parameter value, or using the wrong analytic equations); hold-ups (such as unscheduled system down-time due to a system crash, or unexpected system slow-down due to high system loads); circumstantial errors (such as blotching the conversion to the latest version of an operating system); and "natural" accidents (such as operating system crashes or hardware failure). Mistakes such as these rarely occur intentionally, but they do occur frequently.

Mistakes in computing use is a topic that has surprisingly limited coverage in the literature. What materials there are focus on programming errors (Youngs, 1974), program debugging (Rustin, 1972), or fault-tolerant and reliable software systems (Boehm, 1976). These are generally machine-centered concerns for computing specialists and rational analysts. Though such concerns may be familiar and important to computing specialists, they may be quite remote to many instrumental users. User-centered concerns are often indirectly approached through efforts to improve or "naturalize" the user-system interface (Dzida, et al., 1978; Schneiderman, 1978). Our interest is in examining how users deal with mistakes in computing system use, and the relationship between these mistakes and other work activities.[6]

We saw in the CSRO data that many system bugs (i.e., someone else's miscalculation) are either repaired (if the user can get the system fixed), circumvented with the help of additional software or avoided by a rearrangement of computing tasks. These are all strategies users follow in dealing with system design errors and maintenance inadequacies, and they indicate that even sophisticated computer users may adapt and reshape their work rather than invest the time, skill, and effort necessary to get the system corrected.

Rational analysts seem to suggest that mistakes in computing use can usually be accounted for by the inadequacies of computing resources or in the incompetency of users. But at CSRO, the computing environment is sophisticated and the users highly talented. Structural analysts might show how the uncertainties that users face in using a complex computing system facilitate mistakes. Human relations analysts might focus on whether users are satisfied with the reward system, which encourages good computing use and documentation practices in order to discourage computing mistakes. But interactionist analysts point to (a) the dealings between users and specialists to overcome inadequacies in system documentation, (b) strategies for circumventing regulated access to computing specialists, (c) users' restructuring their work tasks to accommodate emerging system limitations, and (d) user/specialist negotiations over work, resource, and computer use sovereignties.

We have seen how the social dynamics of computing use in organizations affect the efficacy of computing, and how these effects are viewed from the six perspectives. Now, we will move on to examine the consequences of how computing is used in organizational settings.

4.0 IMPACTS OF COMPUTING ON ORGANIZATIONAL LIFE

Analysts of almost every persuasion have suspected that a technology which enlarges the information processing capacity of people or organizations by orders of magnitude must have potent influences on their interaction and work techniques. What kinds of differences do computerized information systems make in the nature of the activities performed by and within organizations? Most of the accounts of the impacts of computing on organizational life focus on the ways in which computing alters the efficiency or effectiveness of organizations, the ways in which decisions are made, the work life of participants in the organizations, the ways activities are structured, the kinds of control managers can exercise in their administrative domains, and the power of different participants to influence the activities of their organizations. Different analysts who adopt different perspectives approach the questions "What difference does computing make?" or "Is computing a good organizational technology?" very differently (see Table 1). The main line of development in this section is to examine the consequences of organizational computing use as viewed by analysts from each of the six perspectives.

Rather than examine every important difference that computer use may make in organizational life, we will examine a few important areas which have been most studied. These include the impacts of computer use on the way decisions are made in organizations, the worklife of computer users, and the alteration in patterns of power and managerial control. We will not examine other topics, such as the ways in which computer use may alter the "structure" of organizations, the ethos of its members, or relations between organizations.

4.1 Decision-making

A critical question faced by anyone who analyzes the role of automated information systems in decision-making is the directness of the linkage of data, information, the way decisions are made, and the effects of different decision styles on activities within and by an organization. This section will examine this question from several viewpoints. Then several classes of automated information systems will be examined. Analysts who focus on the role of computer-based systems in supporting decisions usually differentiate among different kinds of tasks which are usually performed at different hierarchical levels in organizations. Most common is the trichotomy of "operations," "management-control," and "strategic planning" (Burch, Strater, and Grudnitski, 1979). The lion's share of computing in organizations is devoted to applications which support routine operations within the organization: billing, transaction processing, simple record-keeping, engineering analysis, etc. Fewer computational resources are devoted to applications which support the ability of managers to schedule, allocate, and control their resources. And the applications used to support long-range planning and policy analysis are the most analytically interesting and the rarest of all.

4.1.1 Data, Information, Automation, And Decisions -

Rational analysts take the links between data and decisions to be direct, while analysts from other perspectives treat them as more problematic. H.A. Simon has clearly been the most influential theorist and structural analyst to link computer-based systems and decision-making in organizations. The conjunction of computing and decision-making is natural for Simon, since he has developed a view of organizations which emphasizes people's activities as decision-makers and problem-solvers (Simon, 1947, 1973, 1965, 1977). He argues that data and methods which help focus attention and evaluate choices improve the technical performance of a decision-maker. He has also argued that computer-based information systems can help participants in organizations act more rationally by enabling them to compensate for weaknesses in judiciously selecting and remembering information (Simon, 1973, 1977). To the extent that computer-based systems can help them organize and filter larger volumes of information and that simulations enable them to consider a wider variety of complex dynamics simultaneously, Simon views computer-based systems as helpful instruments. Simon's claims have rarely been tested empirically, but they have provided the framework for much theorizing about the social roles of computer-based information by analysts who adopt a rational perspective (Burch, Strater, and Grudnitski, 1979; Licklider and Vezza, 1978; Streeter, 1974).

Many studies which relate computer-based information systems to decisions made by actors in organizations assume that there is usually a good match between the data available in computer systems and the kinds of information used to inform decisions (Burch, Strater, and Grudnitski, 1979; Kanter, 1977; Simon, 1977). In practice, the relation between the data contained in computerized information systems and the decisions that they are held to inform is problematic. No one is suprised when the connections are direct and simple. For instance, most readers will have had the experience of altering airline flights in mid-trip with the assistance of an airline reservationist who uses a computerized information system to find available flights. And when we are told that a police patrolman is able to locate more stolen vehicles with the assistance of an on-line stolen vehicles file which he can access through the department's dispatcher, our expectations that information in computerized systems is useful are confirmed. These examples typify many cases in which some participant in an organization directly uses some item of information from a computerized information system to inform a decision. Clearly, decisions based upon data from computerized systems might be poor if the link is direct between data and decision, and the data used is in error. There are also several kinds of situations in which the link between data and decisions is not so direct.

Rational analysts readily identify swollen data bases as troublesome. Many automated information systems have been built which provide data that is unusable to the people it is supposed to inform (Ackoff, 1967; Danziger, 1977; Dery, 1977). The problems vary, but they include systems which contain inappropriate data items, data which cannot be cross-referenced in useful ways, data which is inaccurate or out-of-date, and reporting formats which are cumbersome. Often these difficulties appear amenable to social or technical fixes.

Structural analysts also identify situations in which the decisions are made independently of the formal data system. Gibson (1975) reports an interesting study of the ways in which decisions to find new branches for acquisition were made by the staff of an Eastern Bank. A management analyst who computed the costs and potential profitability of prospective acquisitions believed that his analyses were the primary elements in decisions about which banks to acquire. However, his supervisor, who was responsible for acquiring new banks, used a variety of informal sources such as friends and newspaper stories to locate prospective branches.

In one land development firm we have studied, financial analysts prepared extensive cash-flow and profitability analyses for new development projects. According to the financial analysts, the reports were critical aids for managers of new development projects. However, according to the project managers, the reports were of minor utility since any of the projects they proposed were profitable in the climate of land development they faced. They were more concerned with shaping a project so that it would meet with the approval of the local planning commissions. Project managers spent more of their time coping with regulatory uncertainty than with fiscal uncertainty, and they evaluated their reports accordingly.

Several years ago we studied the use of a manpower allocation model which was developed by a major Mid-western police department. The use of the model received national attention, and we expected to find it of real utility to police officials who were responsible for shifting allocations between different police precincts. When asked about its usefulness, the police captain responsible for staffing replied, "It's really useful, if you want to allocate manpower." The police staff had developed the model in the late 60's with the assistance of analytical support from a major computer vendor. In 1969, the police were able to increase their staffing by 30% by convincing the city council that ghetto riots and increasing crime rates were imminent. The model was used to help decide how the new recruits were to be allocated to the various precincts. In the early 70's, the riots didn't materialize, the crime rate fell, and no new police were hired. The original allocations to different precincts were maintained, and the model wasn't needed.

Both rational and structural analysts focus "inward" on the formal task organization and on the decisions that are most critically related to the formal tasks. Interactionist and organizational politics analysts examine the web of social and economic relations in which each social unit is embedded. These choices have profound effects on the sense one makes of computing use. For example, a rational analyst would argue that computer use would be of little value to WESCO, discussed in section 2.1, if it did not help engineers design products better or more easily. In contrast, a political analyst would argue that automated design programs could be of tremendous value in helping engineers convince auditors of their design choices, even if the automated system did not alter the quality or lower the direct cost of engineering design.

Rational and structural accounts of computing in organizations emphasize the "formal tasks" of organizational participants (Burch, Strater, and Grudnitski, 1979; Galbraith, 1977; Kanter, 1977; Simon, 1977). These include engineers designing, accountants calculating rates of return and balancing books, and clerks auditing and correcting records. Interactionist analyses also examine the work setting of participants in organizations and examine the way they adopt, define, and negotiate all of the tasks they find important during their working day. People attend to their ongoing social relations in their work groups, their relationships with clients and staff in other organizational units, with auditors, and with subordinates and superordinates in the organization. These "decisions" directing attention may focus on ways of making sense of new activities in one's immediate setting, in maintaining autonomy or security in the work group, in hiding unauthorized work practices, and in maintaining the cooperation of the variety of people with whom one interacts. As we shall see later, these decisions, which are not "about" the public content of a participant's authorized task, may nevertheless play important roles in shaping the uses of computing in organizations.

But this broader ecology of social relationships in which organizational members act can also influence the relations between computer use and decisions. Computing can serve as a symbolic element in organizational life. In the cases of Riverville described in section 1.2 and of WESCO described in section 2.1, computing was used because it convinced important parties that careful decisions were being made. In the case of Riverville, UMIS was not used to alter the decisions made by managers in the municipal welfare agency. In the case of WESCO, we described an episode in which engineering calculations were shifted from paper and pencil to computers to provide the appearance of greater accuracy. These are extreme cases in which computer use serves almost exclusively symbolic and political functions. However, more common examples are those in which computer use serves several social roles simultaneously. Computing may be used to help a manager plan his organization's activities _and_ gain more credibility among other managers for his plans (Keen and Scott Morton, 1978).

4.1.2 Operations -

Routine operations such as billing and airline reservation processing were thought to be relatively dull. If there is any class of task for which computer systems ought to pay off, it is the support of routine operations. Most of the system successes reported by managers or designers of computer-based systems were written from a rational perspective. The typical report describes how goals were set and how the author and his coworkers successfully designed a computer-based system to meet them. All the drama focuses upon the battles faced by the implementers in getting a "successful" system designed on time, within budget, and loved by its users.

There are relatively few careful studies of computer use in organizations and few of these focus on routine operations. The better evaluations are the less triumphant and report complex or ambiguous evaluations. Many routine systems such as traffic ticket processing help increase organizational efficiency (Colton, 1978b; Kraemer, _et al_, forthcoming). Systems may fail because they are technically unsound (e.g., response time is too slow in a demanding decision-making environment) or because they don't contain terribly useful or accurate information (Kling, 1978b). Most important, the criteria adopted for success may strongly influence one's evaluations. Laudon (1974), Colton (1978b), and Kraemer, _et al_ (forthcoming) have all studied police patrol support systems which provide information about wanted persons, stolen property and criminal records to patrol officers in the field. Colton, and Kraemer, et al examined the use of these systems by patrolmen and found the systems to be "successful" with respect to two measures. Colton contrasted a "successful" system in which the mean reponse time is 5-10 seconds with a nearly worthless system in which the mean response time was ten minutes. Patrolmen made about 4 times as many inquiries, per capita, with the better automated system. Kraemer, et al found that police who used a locally automated information system were much more likley to find people with outstanding arrest warrants and locate stolen vehicles than were police who only had access to statewide and national systems. These are the kinds of internal efficiences which rational analysts identify as major values of computing use.

However, as one enlarges the array of activities and subjects included in an evaluation, the picture can alter dramatically. For example, Colton also reported that the (ex)Chief of the Kansas City Police Department at times lamented the enhanced efficiency of Kansas City's system in helping patrolmen find stolen cars, unpaid parking tickets, and unregistered vehicles, because patrol officers made more field stops for these relatively minor offenses and displaced time from other important police tasks. Laudon (1974) adopted a frame of reference that included the network of criminal justice agencies from police through courts. He argued that the increasing success of police in locating stolen vehicles and in citing minor traffic offenders further clogged the courts, which were already jammed. Furthermore, Kraemer, et al found that in locally automated sites police were more likely to detain people who should not have been detained and to arrest people who should not have been arrested. Studies which identify an array of parties with possibly conflicting interests in computing support for routine operations are relatively rare. Sterling's (1979) study of the frequency of billing errors attributed to automated systems is an example.

Again, systems which may perform well according to efficiency criteria narrow in scope (e.g., cost savings for the organization) may be problematic when viewed more broadly (e.g., inconvenience to clients). But it is also true that computerized systems which seem ineffective when evaluated with respect to narrow criteria of internal efficiency may appear of substantial value to their users when they are viewed in the context of a larger ecology of social relations. UMIS did little to improve the internal efficiencies of the welfare agencies in Riverville, but it improved their image of efficient administration enough that they had an easier time attracting Federal funding. The engineering simulations at WESCO which confounded the designers but which helped move a stalled project through another stage of auditing were valued for their political role.

Many computer-based systems persist because they save time, money, or tedium for their primary users. Some of these systems may pose a few important problems for their users or other relevant parties (e.g., clients, data subjects, colleagues). The use of automated text processing at CSRO which we reported in Section 2.10 illustrates such a case. As the groups effected by the use of a computer-based system increase in number and variety, one can expect that computing will serve some groups better than others. Moreover, it is also more likely that some groups will use computer-based technologies to alter their power and dependency relative to other groups rather than simply using the technology to maximize efficiencies in their own operations. Efficiency gains should be <u>findings</u>, however, rather than assumptions.

Rational analysts usually evaluate computerized systems in terms of efficiency or effectiveness criteria which are referenced to the computer-using organization or one of its subunits. In contrast, political analysts assume that different groups which use a computer system or which interact with a computer-using group may have conflicting orientations and interests. In empirical inquiries, a political analyst would examine the orientations and interests of a wide array of participants in a computing milieu rather than assume they were relatively homogeneous. Thus, a political analysis of an organization's milieu will be much more informative for understanding the utility and effects of computing on internal operations.

4.1.3 Management Control -

Control systems are usually initiated by or for managers to help them direct their organizational resources or the activities of their subordinates. These systems parallel lines of authority and usually measure only a few aspects of behavior or activity. Rational analysts have argued that computer-based information systems can dramatically increase the effective control of higher level managers. Automated control systems are seen to provide managers with fine grained, timely, and accurate information about the activities within their administrative domains. But few studies have empirically investigated the accuracy of these claims.

Independent of automation, organizational control systems may be problematic. When the resource which is controlled is inanimate, like parts for a manufacturing plant, the greatest dilemma may be in developing a good control procedure. Rational analysts have observed that a poor procedure, independent of whether it is automated, may lead to costly overstock (Ackoff, 1967). Conversely, a good procedure, such as "material requirements planning," may demand such fine-grained records--of sub-assemblies, sub-sub-assemblies, their constituent parts, and stock levels to be manipulated--that for complex manufactured items that are made to order, digital computers provide the only economically feasible medium.

When the activities which are to be controlled depend critically upon people's skill and cooperation, human relations theorists and political analysts have noted that control systems can become even more problematic (Lawler and Rhode, 1976). The staff of an organization may have several conflicting goals, but most control or reporting systems are designed to measure one or two aspects of a person's or work group's performance. These control systems are easily "gamed." The participants in an organization may simply alter their work style to "make the numbers look good." Assemblers may produce more components, but they may be more components of lower quality. Employment counselors can gravitate toward clients who are easiest to place in jobs. Furthermore, these shifts may have severe consequences for the organization's achievement of a variety of important goals. Co-workers in assembly may sufficiently intensify their competition that morale drops and production suffers through continual turnover and the costs of retraining. Employment counselors may turn their attention away from the clients most in need. Or they may place unsuitable clients to raise their own performance measures in the short run but lower both the agency's credibility with employers, and the number of potential placements, in the long run. These are dilemmas of organizational control. And these dilemmas are not simply resolved through the use of automated reporting systems.

In studies of computer use, it is often difficult to decide which alterations to attribute to the algorithm or strategy adopted and which alterations to attribute to the <u>automation</u> of the algorithm or strategy. That dilemma is particularly critical in the case of control systems. The keen separation between the algorithm and its automation is often moot in practice because computer-based information systems may be the only technically or economically feasible instrumentality to implement them.

The efficacy of several control systems has been studied in private firms by Markus (1979) and in public agencies by Albrecht (1976), Herbert and Colton (1978), Kling (1978b), and Kraemer, <u>et al</u>(forthcoming). At this time, the efficacy of these systems seems mixed. Albrecht found that a caseload reporting system used in a large metropolitan court led to substantial reductions in case processing time. These findings are consonant with the rational perspective. In contrast, Kling and Markus found control systems which were of little utility to managers in controlling activities within their administrative domains. Markus studied a financial/accounting system and found that its primary utility was to rationalize the existence of the office of the corporate comptroller. Similarly, Kling explained the persistence of UMIS in primarily political terms.

Kraemer, <u>et al</u> investigated budget monitoring systems in local governments, and systems to help police commanders allocate their staff across beats, districts,and shifts. Herbert and Colton (1978) also studied police manpower allocation systems. Kraemer <u>et al</u> found that budget monitoring systems were favorably viewed by central managers as helping them to track the expenditures of operating departments and to predict potential problem areas. The budgetary control prerogative of central managers is so well accepted that the staff of operating departments did not find <u>this</u> surveillance to be intrusive. When the staff in operating departments complained about budgeting systems, it was primarily because the systems were designed to serve the needs of central managers but did not serve <u>their</u> needs very well. Usually these complaints were assuaged when budgeting systems were redesigned to accommodate the interests of staff <u>both</u> in central finance departments and in the operating departments.

However, neither Herbert and Colton nor Kraemer, et al found police manpower control systems to be particularly efficacious. Herbert and Colton present a structural argument for the difficulties of linking automated manpower allocation to improved police performance by invoking several structural arguments. One line of argument addresses computing and police performance as part of a larger strategy for crime deterrence. Thus, manpower allocation models are part of a strategy to increase the the number of criminal apprehensions by increasing the speed with which a patrolman can reach the scene of a crime; and manpower allocation schemes are used to deploy patrolmen to the beats and shifts where apprehensions would be most likely. But, Herbert and Colton argue, even if the models were relatively efficacious in helping police departments allocate their patrol, the effects on criminal activity would be negligible. Even if it were twice as likely that a suspect would be apprehended, his chance of conviction by a court would be only slightly increased given the delays, plea bargaining, and dismissed cases which are common in the American courts. In contrast, the analysis presented by Kraemer, et al lends itself to a political interpretation. They find that manpower allocation models are least efficacious in cities where elected officials have relatively strong influence in computing decisions. In these cities, elected officials are likely to have strong influence in decisions about the allocation of other resources as well, including police. Since the quality of police service is a highly politicized topic in cities with moderate or high crime rates, city councilmen may refuse to cede to model-based analyses which lead to the loss of patrol support in their districts. For example, manpower allocation models in Atlanta indicated that police should be moved from outlying suburban areas to inner city beats. The councilmen in the suburban areas mustered a majority of votes to prevent the proposed shift. Thus, the models were "ineffective."

Examination of control systems for budgets and police manpower reveals interesting contrasts in efficacy. Here a structural perspective seems fruitful. While both expenditure levels and crime rates are subject to exogenous influences, the extent to which departments spend their <u>allocated budgets</u> is more subject to adminstrative control. Moreover, budgetary control has been a traditional prerogative of central managers, who set budgets for operating departments at the beginning of the fiscal year. The staff of operating departments may feel that budget monitoring by central accounting staff doesn't strongly intrude upon their traditional prerogatives.

When there are major conflicts over who will control critical organizational resources, computerized information systems may simply be used as political instruments by the contenders -- and fought as political intrusions. Albrecht (1979) presents an intriguing study of a case-tracking system which the legal staff (e.g., judges) in a Southern court introduced to manage the work of the probation staff. Each of the two groups brought a different orientation to its work with defendants and convicts, and each was able to exercise moderate autonomy in its control over the meaning of its work. Albrecht indicates that the legal staff were concerned that cases be processed through the courts in an orderly manner and emphasized due process. In contrast, the probation staff emphasized rehabilitating individuals so that they could become productive and trusted members of the community. When the information system was being designed, each group proposed a reporting structure which <u>minimized</u> its accountability and <u>maximized</u> the visibility and possible control over the other group. An automated system which included a compromise set of data was built and operated for four years. But it was primarily used as a record-keeping system and rarely used to enhance the control of court administrators. Finally, it was removed and the court reverted to a manual record-keeping system. Albrecht views this information system as an instrument in the struggle between the legal staff and probabtion staff for control of each other. Neither group was able to gain sufficient power to force the other to submit to its form of measurement and mangement. Since no group could tightly manage the other, and thus provide "objective" data about the productivity and efficacy of court activities, the automated system was a sterile tool. This case also highlights the close coupling of management control systems and the exercise of power in organizations. Organizational power will be carefully examined in section 4.2.

4.1.4 Policy-making And Strategic Planning -

The roles of automated information systems in policy-making have been examined from rational, structural, and political perspectives. This section examines sample studies from each framework.

An example of the rational prespective is provided by a recent account of computer-based models by Licklider and Vezza (1978):

> "Computer-based modelling and simulation are applicable to essentially all problem-solving and decision-making...they are far from ubiquitous as computer applications...The trouble at present is that most kinds of modelling and simulation are much more difficult, expensive, and time-consuming than intuitive judgement and are cost-effective only under special conditions that can justify and pay for facilities and expertise."

These casual explanatory comments typify the weaknesses of much rational commentary on the social effects of computer-based technologies. They embody a simplified version of Simon's decision-making perspective, and lack the attention to social context which Simon, on occasion, provides. Neglecting the _obiter-dicta_ claim that modelling and simulation are "applicable to essentially all problem-solving and decision-making," presumably including ethical decisions, one is left with an odd account of the problems of modelling. Models "are far from ubiquitous," and "the trouble is" they are difficult and costly to develop and use. But the appropriateness of modelling is not linked by Licklider and Vezza to any discernable social setting or the interests of its participants. Licklider and Vezza's claims are not aimed at policy-making in particular. They could include simulations for engineering design as well as for projecting the costs of new urban development. However, their comments typify the rational perspective when it is applied to information systems in policy-making; the presumption is that differences in social settings make no difference.[7] But scholars who have carefully studied the conditions under which computer-based technologies are adopted and used have found that the character of social settings is a potent influence (Danziger and Dutton, 1977; Greenberger, _et al_ 1976; Kling, 1974,1978a,b; Rule, 1974). Whether models and simulations "are applicable to essentially all problem-solving and decision-making" by virtue of helping inform decisions, or whether they are used to rationalize and obscure the bases of decisions made on other grounds, depends in part on the _degree of consensus_ over means and ends held by the active participants in a decision arena (Greenberger, _et al_ 1976; Kling, 1978a). Exactly what "special conditions" Licklider and Vezza believe "can justify and pay for facilities and expertise" cannot be discerned from their account. The reader is provided with an ambiguous analysis, which is couched in rational rhetoric, but which can be interpreted within broad political understandings of organizational action.

Greenberger, Crenson, and Crissey's (1976) investigation of the use of computer-based modelling systems as guides to policy-makers in public agencies is primarily a structural analysis with some political elements. They studied the role of econometric models in developing U.S. fiscal policy; Forrester's WORLD III model, which stimulated the "limits to growth" debate in the United States; and an operations research model for locating fire stations used by the City of New York. An advocate of rational modelling would assume that models such as these could and should be used by policy-makers to help sharpen their perceptions and help select among alternative policies. Greenberger, Crenson, and Crissey rarely found that policy-makers' choices were influenced directly by model-based analyses. Political actors often used models, but they were employed to generate support for policies selected in advance. When modelling efforts were influential in a decision arena, it was often the _modeller_ who was called upon to inform decisions rather than _modelling runs_. Models did help inform decisions indirectly by stirring partisan debate. While "modelling as advocacy" was best illustrated through the "limits to growth debate" and through their account of Laffer's predictions of the U.S. GNP in 1971, modelling as a tool of advocacy is commonplace. Under such conditions, the most constructive role for models has been to help advocates of different positions make their cases and to critique the assumptions of their antagonists.

Greenberger, Crenson, and Crissey's subtle analyses of each modelling effort indicates that details and structural arrangements differ across problem domains and social arenas. Bargaining over fire-house locations in New York City differs from predictions of the Gross National Product in response to an increase in the Federal Reserve Board's prime interest rate. City councils and Congress, firemen's unions and bankers, act differently. But in each case, Greenberger, Crenson, and Crissey take the world of policy-making as given, with its short time horizons, fluctuating attention to issues, and attempts of participants to mobilize support, placate critics, and displace problems that don't require immediate attention. They find that modelling does not easily fit the fragmented world of public policy-making. They note that modelling efforts often require good definitions of the questions to be asked, while policy-makers are often working with shifting definitions of the dilemmas they face. In addition, modelling efforts often require several years to design, program, and fine-tune. Many policy matters are resolved more rapidly; indeed, many political actors are out of office or transferred to other jobs within any two-year period. Greenberger, Crenson, and Crissey also note that models help an actor gain intellectual mastery over a given problem domain by integrating many factors and their interactions. In contrast, political arenas, particularly in the Federal government, are designed to manage problems by factoring them into chunks which can be delegated to different administrative and regulatory agencies. Each of these structural features (e.g., time horizon, stability of people and problems) becomes an element in their explanation about why modelling has been problematic. In addition, they play down the political attractiveness which modelling and modellers offer politicians. If the modeller is called upon to provide sage advice, he can be viewed as a flexible political resource. He can be chosen to have a world view which is closely aligned with the politician's, but to use the "objective authority" of his modelling to enhance the credibility of casual, partisan analyses.

Despite the structural "misfits" between model-based analyses and the dynamics of policy-making which they emphasize, Greenberger, Crenson, and Crissey are optimistic about the value of modelling efforts and recommend strategies to improve their utility. They focus upon social strategies that institutionalize the development of specific models which may be used to answer recurrent questions and to intensify the uses of countermodelling and multimodelling. They clearly eschew those explanations of the failures of rational modelling efforts which hinge on the technical weaknesses of contemporary models.

In a study of the use of automated information systems in municipal policy-making, Kling (1978f) contrasted the explanatory power of rational models and organizational politics models. Automated information systems that provided demographic, economic, housing, and transportation data were available to most of the city councils, planning staff, and top administrators in the 42 cities he studied. Computer-based reports were provided to city council several times a year, on the average. In about half the cities these reports never led to clearer perceptions or surprises about city conditions. In less than 10% of the cities did these reports generally provide clearer perceptions or surprises to city council members. However, in 35% of the cities, the reports were generally used to enhance the legitimacy of decisions and perceptions of city councilmen. In about 10% of the cities, the reports were generally used to gain publicity for programs. In about 25% of the cities, these reports were sometimes used to legitimize perceptions and gain publicity for programs supported by council members.

These patterns could be attributed to the primitive state of automation in American cities. If cities utilized more sophisticated data resources, then perhaps policy-makers would find more surprises in the reports they received and be less likely to use them as political resources. Kling developed several measures of the degree and richness of automated data systems supporting policy analysis in each of the 42 cities. He found that in more highly automated cities, policy-makers were more likely to report clearer perceptions and surprises gleaned from reports based on automated analyses. However, he also found that policy-makers in more highly automated cities were more likely to use these same reports to legitimize their perceptions and gain publicity for their preferred programs. These findings indicate that computer-based analyses are a social resource used by political actors in the same manner as other social resources. They don't alter the policy-making style of political bodies; they are appropriated by policy-makers and adapted to <u>their</u> styles of organizational work.

These studies indicate that computing can play multiple social roles in organizations which adopt them. While the studies cited here have been carried out within public agencies, there is no reason to believe that the <u>internal organizational dynamics</u> of private organizations are substantially different. For example, Pettigrew's (1973) study of computing adoption decisions in a private manufacturing firm indicates that office politics can be as pervasive as that in public agencies. In an effort to implement an analytical model to assist the officers of a Boston bank in locating new branches, Gibson (1975) also discovered actively "political" modes of decision-making at the higher corporate levels. Simon indicates similar perceptions when he suggests that the manager in a more automated firm:

> "will find himself dealing more than in the past with a well-structured system whose problems have to be diagnosed and corrected objectively and analytically, and less with unpredictable and sometimes recalcitrant people who have to be persuaded, prodded, rewarded, and cajoled.... Man does not generally work well with his fellow man in relations saturated with arbitrary authority and dependence, with control and subordination, even though these have been the predominant human relations in such settings in the past. He works much better when he works with his fellow man in coping with an objective, understandable, external environment (Simon, 1977: pp. 132-133).

Despite this hope that automation will help diminish the abrasive conflicts within organizations, there is no evidence that computer-based systems have "depoliticized" the organizational settings in which they are employed. Strassman (1973) observes:

> "The motivating forces behind every game in the systems field are the achievement of power, influence, 'political posture,' high-compensation levels for its members, and a disproportionate share of the corporation's resources."

There is a vast literature on the design of information systems for organizations, but remarkably few studies actually evaluate the use of systems in place. And only a handful of these treat the interplay between automated information systems and the social order of their host organizations.[8] The studies which have been summarized here indicate that computer-based information systems that are regularly used are grafted into the ongoing politics of their host organizations. Rather than altering the character of social "relations saturated with 'arbitrary' authority and dependence," automated systems are best received when they buttress the positions of organizational actors with substantial power (Kraemer and Dutton, 1978; Kling, 1974; 1978f; Kraemer and Dutton, 1977).

4.2 Organizational Power

In section 4.1.3 we discussed "managerial control," by which an administrator increases his influence within a domain of activity over which he has primary supervisory responsibility. However, there are many arenas of organizational life in which participants share their influence, and may attempt to increase their influence. One participant exerts power over others to the extent that he can effectively influence them to see things and do things his way. It is a relatively common observation that computer-based systems increase the influence of parties who have access to tne technology and understand its use (Danziger, 1977). Such an analysis led Downs (1967) to suggest that data custodians will gain power relative to other staff, and that full-time administrators will gain power relative to elected officials.

Recently, Kling (1978f) and Kraemer and Dutton (1979) have investigated the role of information systems in altering balances of power in American municipal governments. They collected data for investigatiing alterations in power relations attributable to the use of computer-based reports. In each city, the researchers coded the extent to which different actors gained or lost power because of computer-based reports. "Power" was treated operationally, as "effective influence;" actors who seemed to have gained influence in the outcomes of decisions made during the previous year or two were viewed as having gained power. A weak group gaining power need not be relatively strong, and a strong group losing power need not be relatively weak.

The following patterns stood out in the data:

1. In a majority of the cities, there were discernible shifts of power to or from some role (e.g., mayor) attributable to computer use.

2. However, there were no shifts of power to or from any single role (e.g., mayor) in a majority of the cities. In each city, different sets of actors gained or lost power because of computer use. Moreover, many actors had their relatively power unaffected by computer use. (Respondents also attributed shifts of power to a variety of sources including demographic changes in the city, personal style of top officials, etc.)

3. Data custodians (e.g., planners) were most likely to gain power, and never lost power, because of computer-based analyses. Nevertheless, data custodians remained relatively weak. At best they appeared as the favored experts of more powerful actors. At worst, their council was distrusted and their reports receive little sustained interest.

4. Supporting Downs, city councilmen were most likely to lose power (20%) and rarely gained power (5%) because of computer-based analyses.

5. Supporting Downs, top-level administrators (city managers and CAO's) often gained power (27%) and rarely lost (3%) power when there were any shifts at all.

The conditions under which mayors and departments gained and lost power were more complex. The top elected officials in a city were more likely to gain power and less likely to lose power than all other role-takers except the data custodians. This suggests that computer-based analyses <u>reinforce</u> the patterns of influence in municipal governments (Kling, 1978f; Kraemer and Dutton, 1979). Kling (1974) suggests that computer-based information systems should reinforce the structure of power in an organization simply because computer-based systems are expensive to develop and use. Thus, top officials who can authorize large expenditures will, on the average, ensure that expensive analyses serve their interests.

However, the data reported by Kling and Kraemer and Dutton indicate even more subtle patterns of influence. Correlations were computed between power shifts attributable to computing and measures of both city size and the extent of automation in policy analysis. Departments gained power while top officials lost power in the <u>larger</u> cities. This pattern follows the structure of influence in American municipalities. In the smaller cities, the mayors and city managers were able to stay on top of department operations and place a strong stamp upon municipal activities. In larger cities, the departments could be vast fiefdoms (particularly police and public works) and the top officials were placed in a weaker bargaining position vis-a-vis department heads. In the cities with several hundred thousand residents the larger departments had sufficiently large budgets to afford their own skilled analysts to build information systems that suited their needs, and do it with less scrutiny from top officials. Thus, in these cities, it is not top officials who gained most, but managers who were sufficiently high in the organizational hierarchy that they could command large resources.

A general observation drawn from the analysis of computer-based systems for policy analysis in cities is that they often serve as _political_, _power reinforcing_ instruments (Kling, 1978f; Kraemer and Dutton, 1979). They differ from automated upward reporting systems which may cause data providers to lose power to data collectors (Kling, 1974). In general, to predict the influences of a computer-based information system, one must have in mind a sharp characterization of the distribution of influence in the social setting in which it is used (Dutton and Kraemer, 1978; Gibson, 1975; Greenberger, _et al_ 1976). Automated information systems should be viewed as social resources which are absorbed into ongoing organizational games, but which do not materially influence the structure of the games being played. In organizations in which "normal business" is highly politicized, computing will not be a neutral resource. It can easily be used to reinforce the power of potent actors.

This analysis is neither vacuous nor obvious. Recently, McLaughlin (1978) recommended that the Office of the President should develop a "policy management system" with extensive automated support to help provide greater control over the (then) U.S. Department of Health, Education and Welfare. Since the Federal cabinet agencies have developed into extensive bases of independent power vis-a-vis the Presidency, they may be analogous to departments in the larger cities. The analysis presented here suggests that extensive automation is likely to reinforce the power of the executive agencies. In that case, McLaughlin's strategy would backfire.

This analysis indicates that the political order of the social setting in which a computer-based system is utilized must be well understood, in addition to the technical features of the system, to predict the likely uses and impacts of computing. This principle undermines the sufficiency of the formulations of analysts who emphasize the technical characteristics of systems and neglect the political dynamics of the settings in which automated data systems are utilized.

4.3 Computing And Work

There is no paucity of images to portray the effects of computing on work life. Bell (1973) and Myers (1970) provide enthusiastic and largely rational accounts of computing as an aid to "knowledge work." They both emphasize how computer-based technologies enlarge the range and speed at which data is available. Myers, for example, concludes that, "computing will help relieve 'specialists and professionals' of the time-consuming and repetitive parts of their work." In contrast, Braverman (1974), a class politics analyst, argues that managers conceive of workers as general purpose machines which they operate. He views automation as a managerial strategy to replace unreliable and finicky "human machines" with more reliable, more productive, and less self-interested mechanical or electronic machines. He argues that computerized information systems usually are organized to routinize white-collar work and weaken the power of lower level participants in an organization. Human relations analysts occupy a middle ground between these extremes of euphoria and gloom. They believe that the effects of computerization are contingent upon the technical arrangements chosen and the way they are introduced into the workplace (Argyris, 1971; Kling, 1973). While human relations analysts place considerable value on jobs which allow people opportunity for self-expression and trust (Argyris, 1973); they argue that the character of jobs and the role of computing in work is to be emprically determined (Mann and Williams, 1960; Kling 1973).

4.3.1 Computing And Job Characteristics -

Empirical studies of computing in the workplace have been dominated by human relations approaches (Mann and Williams, 1960; Mumford and Banks, 1967; Kling, 1978e; Robey, 1979). Class politics analysts often draw upon studies done by others, and then interpret them in their own framework (Braverman, 1974; Mowshowitz, 1976), although Noble (1978) has devloped a provocative, empirically grounded account of computer numerical control in machine shops which is based upon his own field studies. Rational analysts usually emphasize the information processing potentials of computing, but remain relatively remote from empirical inquires.

Regardless of their theoretical orientation, most analysts view computing as a potent technology which is likley to have a powerful influence on the workplaces where it is used. It is easy to think of technological changes, such as the transformation of craft fabrication to assembly lines, which have been organized to have powerful influences on the character of work. Kling (1978e) recently investigated the impacts of computer use on the character of white-collar jobs through a survey of 1200 managers, data analysts (e.g., urban planners), and clerks in 42 municipal governments. While his study is carried out within the human relations tradition, it casts doubt on the easy assumption one finds in class politics and rational analyses, that computing is a potent workplace-transforming technology. Most of Kling's respondents used computer-based reports and attributed job-enlarging influences to computer use. Respondents also attributed increases in job pressure to computing. These effects increased with the extent to which respondents used computer-based reports in their work. Computer use had perceptible, but <u>not dominant</u> effects on the jobs of many people who used the technology; overall, Kling's respondents reported that computer use enlarged their jobs, but did not profoundly alter the character of their jobs.

Computing was viewed as a salient technology by managers, data analysts, and clerks who utilized it regularly. For some, computing substantially increased the ease with which they could obtain valued information, and made possible more complex data analyses. For others, it had little utility. White collar workers in several different occupational specialties attributed clear, often positive influences to computing in their jobs. Computing use also moderately increased the level of task significance and job pressure in the work of computer users.

Kling's (1978e) findings support those of Whisler (1970) who indicates that managers are better served by computing than are clerks. But even traffic clerks in Kling's study generally attributed job-enlarging influences to computing. Kling's data lends no support to analysts like Braverman (1974) and Mowshowitz (1976), who argue that the dominant effect of white-collar computer use is to diminish the quality of working life. On the contrary, it supports claims that computer use often enlarges the jobs of workers who use the technology.

The detailed patterns and levels of the impact of computing vary from one occupational specialty to another and can be understood best in terms of different work contingencies and information system designs. In addition, the technology is not always easily implemented -- difficulties in getting data, dealing with computer specialists, the computing services organization, and the technology itself all add minor and continual turmoil to the workplace; but overall, the technology does not dramatically change the character of work. Rather, according to Kling's findings, it has a benign and minor influence on the work of the computer users.

4.3.2 Automated Information Systems And Supervision -

It is often assumed that when automated information systems become available, managers and line supervisors exploit them to enhance their own control over different resources, particularly their subordinates activities (Downs, 1967; Whisler, 1970). Two strategies are commonly adopted: reorganizing work so that tasks are more finely subdivided or control over key decisions moves "up" the organization (Noble 1978), or directly employing automated information systems to enhance the grain and comprehensivness, and speed of organizational control systems (Kling, 1974; Lawler and Rhode, 1976). In this section, we examine the latter strategy.

Episodes illustrating the use of automated information systems to better monitor the activities of employees, particularly the quantity of their work, can be found in many computer-using organizations (Kling, 1974). Both the image of computing as enhancing the rationalization of white-collar work (Braverman, 1974; Mowshowitz, 1976), and reports by managers that they were losing autonomy (Mumford and Banks, 1967) support the hunch that such uses are widespread and effective. In short, computing and closer supervision are held to go hand in hand.

Nevertheless, there are few good studies of automated information systems and patterns of supervision. In section 4.1.3, we reviewed the case reported by Albrecht (1979) in which the probation staff of a court were able to prevent the legal staff from employing an automated information ssystem to monitor their casework. Albrecht's case provides a rich account of how a specific information system was introduced with by actors (legal staff) who wished to control others in their work organization. How commonly do organizational actors attempt to utilize information systems as direct aids to supervsion, and how successful are they? Kling's (1978e) study of the role of computing in white-collar work provides some helpful data. Each of his respondents (described in section 4.3.1) was asked to report whether or not automated information systems had increased or decreased the extent to which he or she was supervized. Few of the respondents attributed increases in their level of supervision to computer use. And, Kling found that traffic clerks and managers, rather than accountants, detectives, and planners, most frequently (12%-16%) reported increases in their level of supervision.

The respondents were also asked whether computing provided their supervisors with information about the quantity and quality of their work. Between 25% and 36% of the respondents believed that computer-based reports conveyed such information. In addition, the managers were asked whether computer-based reports helped them monitor and control the work groups within their organizational domains. A majority of the managers indicated either that they received no relevant computer-based data, or that the computer-based data they received were of no use for enhancing their control over their subordinates. A small fraction indicated that computer-based reports were a major aid. And many managers indicated that computer-based data were sometimes, but not usually helpful.

Again, understanding the contingencies of different jobs and the ways in which information systems are used helps us understand these patterns. Computer-based reports are sometimes used to monitor the workloads of detectives, property appraisers, and social workers. However, such data is usually available in manual records as well and is simply spun off as a by-product of the computer-based systems (e.g., a welfare client-tracking system). First-line supervisors often have some concept of the relative quantity and quality of the work done by their staffs. In most police detective details, cases are assigned to individual detectives by a supervisor; planners are assigned work on particular projects and their reports are reviewed by higher level staff; accountants work on particular projects or with particular funds. Much of the work is carried out in small groups where supervision can be informal and close. Even when formal statistics may help, their explicit use seems to produce resentment, suspicion, and controversy over the meaning of the data.

The information systems used by most of Kling's non-clerical respondents were not well tailored to provide information about their work style, or the quantity or quality of their work. Ledgers do not record the activity of an accountant, and demographic data bases do not record the activities of planners who use them.

As we indicated in section 4.1.3, automated systems can enhance the control of one group over another (Kling, 1974).[9] But, in many workplaces where computing is used to automate records and data analysis, the technology has not been exploited by managers tp incresae their information about the activities of their subordinates. Rather, the scenario of computing and control is of accountants in central budget units monitoring the expenditures of line departments; detectives seeking to catch lawbreakers; and municipal staff using automated traffic ticket processing systems to help catch scofflaws. Taken together with Albrecht's study, these data indicate that computing should not be viewed as a technology with necessary, fixed patterns of use and consequences. Computing is selectively exploited, as one strategy among many, for organizing work and information. however, the patterns of computer use appear to fit the workplace politics of the computer-using organization. In current practice, semi-professionals do not attribute increased supervision to the use of automated information systems. Structural analysts would suggest that managers might find other strategies for controlling their subrodinates to be be more effective. Political analysts would suggest that semi-professionals such as probabtion officiers, accountants, and detectives have sufficient control over their own work that managers cannot succeed even when they attempt to use formal reporting systems to enhance their supervisory control.

4.3.3 The Integration Of Computing In Work -

The published studies of computing in the workplace are relatively mechanistic in conception. When workers have little discretion over their use of computing, then they may experience computing as an external force to which they must adapt. However, many white-collar workers have some discretion over the conditions and occassions under which they will use computing.

Detectives, for example, can initiate searches in extensive file systems relatively early or late in working a case. Individual detectives have substantial flexibility in integrating computing use into their work. Similar observations apply to the engineers at WESCO described in Section 2.1 or managers of stock portfolios described by Keen and Scott-Morton (1978). People who have substantial discretion in their decisions about whether, when, and how to use computing can utilize it as a relatively flexible resource to fit many social agendas. It may be used primarily as a rational instrument to save time or to ease the handling of complex bodies of information. Some modes of automation, such as automated testing of electronic compondents, may help managers employ less skilled workers to reduce costs or maintain stable supplies of labor. But its use may also be invoked as a symbolic gesture to appear innovative, as a medium for playful work, or as a political instrumentality. Little is known about the way that semi-professional workers can adapt computing to their worklives, or the ways in which their worklives are altered with different computing arrangements. We believe that the interactionist approach which empahsizes the ways in which people construct social meanings around the different elements of their worklives--tasks, technologies, and people--may provide the richest starting point for this line of inquiry.

5.0 CONCLUSIONS

In this chapter we have analyzed the development, use, and consequences, of computing from six different theoretical perspectives which we introduced in Section 1.3. This has been a rather complex task, even though we have neglected many important aspects of computing in organizations. We have tried to demonstrate how each "phase" of computing--its development, use, and consequeces--entails a complex set of social activities. In addition, we have indicated how the assumptions one makes about behavior in organizations colors what we look for, what we find, and what sense we make of computing developments. We have examined some of the more important and recent studies which shed light on the social dimension of computing in organizations.

The research studies of the few topics we have covered, however, form a rather fragmented and not terribly cumulative literature. Rather than summarize the substantive findings here, we reexamine the six perspectives which we have used to integrate the diverse topics we have examined. Then we suggest some promising lines for further investigation.

5.1 The Six Perspectives In Perspective

The six theoretical perspectives help us understand the assumptions behind the questions we have asked and the answers different analysts have found. The rational perspective dominates the majority of analyses of computing, particularly those which are written by practitioners and found in trade journals or the internal documents of organizations. Rational analyses provide coherent "first approximations" which help provide the reader with simple, memorable, and clear conceptions how computing systems are developed, used, and what interests they serve. Unfortunately, all too often rational analyses sytematically mislead. The rational perspective severely simplifies the world of computing by assuming that major participants in a computing setting share common goals and values. In addition, rational analysts usually assume that the formal structure of authority in an organization provides a good guide to the way activities are carried out. Reference to consensus over goals helps mobilize diverse groups to share, or at least not impede, a collective effort. But statements born of institutional convenience are unlikely to provide adequate conceptual schemes for undertaking serious analytical work. Unfortunately, aside from the most careful studies of computing use which "triangulate" accounts of different participants and examine formal data, much of what we believe about computing in organizational life is based upon a fragmented set of interested accounts.

Structural analyses often enrich the rational perspective by indicating the conditions and arrangements under which specific activities will occur or are appropriate. Human relations analysts add richness to the relatively dry and mechanistic world of structural analysts by "bringing people back in." However, the human relations tradition has been almost exclusively preoccupied with individuals and work groups. They provide managers with studies informing them how best to organize so that their workers will be both happy and productive. However, they provide little insight into other aspects of information technology in organizational life--the diverse incentives for computer use, the fragmented world of computing in many organizations, and the alterations of power between computer users and others. Both interactionist and organizational politics analyses keep people sharply in focus, and are usually less biased towards the interests of managers than are human relations analsysts. They can also investigate a wider array of social forms beyond the task group (Danziger, et al, forthcoming; Kling and Gerson, 1977, 1978). Class analysts have undertaken few studies of computing in organizations. Largely, they treat computing simply as a complex technology which supports the power of dominant classes in a capitalist society. Class analysts have taken the position that work in capitalist society is degrading, and have tried to demonstrate how managers attempt to deskill programmers (Kraft, 1977; Greenbaum, 1976) and clerical computer users (Braverman, 1974). Class analysts play an important role in pointing at the ways in which managers of work organizations often organize work so as to maximize productivity rather than provide richer worklives and significant forms of control for their subordinates.

We believe that each perspective provides us some insight into some questions. For reasons that we have tried to clarify in this chapter, we nevertheless find that the interactionist and organizational politics perspectives provide more analytical bite under a wider variety of circumstances than their competitors. The researcher's choice of perspective leads to many important ramifications for appropriate research methods. If, like rational analysts, one believes that there is consensus over goals, then one "key" informant may provide sufficient data about the development, use, and consequences of a given computer system. On the other hand, if one believes that knowledge about computing is fragmented, that participants are interested rather than disinterested, then one may need many, perhaps several dozen informants, to "adequately" indicate the social contour of a given computer system (Kling, 1978b; Markus, 1979; Scacchi, 1980). These choices have important consequences for the research style most appropriate to the perspective and the questions asked. The more data needed to make sense of a given event or computing arrangement, the more costly the research if many events or systems are to be studied. Conversely, political and interactionist analysts are also more likely to produce qualitative reports since a common research instrument is less likely to capture the array of meanings and nuances communicated by a wider array of situated respondents.

Simply specifying a particular "perspective" is relatively weak since there can be alternative accounts within any approach. For example, Danziger, Dutton, Kling, and Kraemer (forthcoming) examine the explanatory power of three different political models for the control of computing in organizations - elite pluralism, technocratic elite, and reinforcement politics.

Much of the theoretical diversity (or confusion) of the studies reported here reflects the state of organizational theory generally. There are many competing perspectives (see Perrow, 1979). Moreover, scholars of organizations have trouble agreeing on even simple matters such as how to understand "effective" organizations (Goodman, et al, 1977). Studies of the social dynamics of computer use adopt theoretical perspectives from the sociological literatures (e.g., Gerson, 1976; Strauss, 1978a,b), and they reflect important weaknesses or difficulties that sociological theorists have not yet resolved.

5.2 Lines For Development

There is also much that we have left out. First, many computer scientists would have developed this chapter by focusing on specific software development strategies (e.g., structured programming, automated testing), on particular technologies (e.g., minicomputers, networking), or on managerial arrangements (e.g., pricing computer services, centralization vs. decentralization of computing). These are the practical terms with which developers and users of computing often express their choices. Each of the six perspectives developed in this chapter can be used as a frame of reference for analyzing these topics. And we hope that the analysis presented here can be picked up by practitioners and scholars and turned to these more practically framed questions.

Most of the studies which examine computing or computing practices in organizations, emphasize the instrumental side of computing. But computing also is the occassion for symbolic, ceremonial, and ritual behavior as well. For example, "cost-benefit" studies are often invoked as rationales for the adoption of a new computing application or for some major reorganization of computing arrangements. However, cost-benefit studies are often carried out to support beliefs than for their analytical role in helping select cost-effective choices. Important "benefits" such as enhanced organizational control or images of innovation are usually difficult to quantify in monetary terms. As a consequence, "cost-benefit analyses" often boil down to "we knew there would be costs, we hoped there would be benefits." Slogans like "cost-benefit analysis," or "state of the art systems" indicate just a few of the many symbolic and ceremonial events that surround the development and use of computing.

Third, we have treated organizations as relatively closed systems. Recent studies of the industrial and social world in which computing is developed and used indicate that organizations and groups outside the computer-using organization can have profound influences upon computing activity (Kling and Gerson, 1977, 1978; Kling and Scacchi, 1979a,b; Scacchi, 1980). Professional societies provide arenas for their members to learn about kinds of computing which they can "import" into the organizations for whom they work. They are also arenas in which specialists -- accountants, urban planners, production planners, programmers, data processing managers -- develop their conceptions of "adequate" or "interesting" technologies. Programmers now work in a tight labor market in most urban centers, and employers often have to keep their operations technically current to attract and retain skilled staff. These links between computer using organizations and larger social worlds merit serious attention.

Moreover, computer-based information systems are fit into a complex ecology of organizations with increasing frequency. The FBI, state police agencies and police in the largest urban centers operate information systems which contain criminal histories and information about wants, warrants, and stolen property which cross a variety of organizational and juristictional boundaries. Networks of point-of-sale terminals and credit verification networks link geographically dispersed merchants with central banks and credit reporting agencies (Kling, 1978d). Some manufacturing firms which have automated their entering of orders receive streams of automated messages from their larger customers who place orders through their own automated systems. On the consumption side, both retailers and manufacturers are turning to automated inventory control systems which trigger purchase orders in an increasingly automated manner. In short, information systems which are shared between organizations or which contain data about transactions between them appear to be tightening the links between organizations in certain industries. However, the nature of the these developments, their modes of use, and their consequences for their participants are poorly understood. As the number of groups with diverse perspectives who deal with a given computerized system increase, it's character is likely to become more markedly social. Software developed in such settings is likely to be more complex as different groups place somewhat conflicting demands upon the ways a given system should operate and what modes of information porocessing it should incorporate. Moreover, since any system entail some compromises over how it will actually work and what activities it will support, the use of systems of larger "social scope" is even more of an arena for social conflict (Kling, 1978e; Albrecht, 1979; Markus, 1979).

Fourth, an appropriate new line of investigation would directly examine the relations between computer using organizations and their clients. The earliest accounts of computing which examined the losses of personal privacy that accompanied large scale computerization served to focus attention on privacy as an issue. But they also deflected attention from other important elements in the relationships between the public and computer-using organizations. Rule (1974) made an important contribution by framing concerns about personal privacy as an issue about the kind about social control which organizations can exert over their clients.

Sterling's (1979) important study of billing errors opens this line of investigation in a new way. Traditionally, studies of computer-using organizations that provide public services focus on the computer-using organization and its administrative practices (Rule, 1974; Laudon, 1974; Kling, 1978b; Albrecht, 1976, 1979; Kraemer et al, forthcoming). These studies emphasize the character of services, while Sterling's gives explicit attention to the people who are receiving services from the organization.

As a wider variety of organizations automate their operations, the way that computer-using organizations interact with their publics will become more critical (Kling, 1978c,d; Rule, 1974; Sterling, 1979). Moreover, larger firms are more likely to invest in computing than their smaller competitors. In retail sales and banking, for example, these are likely to be organizations whose operations span wider geographical areas and which have larger clienteles. However, larger firms provide a different kind of "service" or product than smaller, local competitors which sell the same commodity. In some cases, such as chain stores, they provide a wider variety of nationally distributed merchandise at lower prices. But they are usually less personal, and less likely to distribute locally produced merchandise. In these cases, the customer finds a wider choice of mass produced items or services, but is likely to have to deal with remote organizational units to make special requests and deal with problems. To the extent that such regional, chain organized firms utilize computer systems to rationalize their marketing, distribution, and billing, computing may appear as "essential" technology. But automated information systems may help support larger scales of effort against which smaller local firms cannot easily compete. Ceteris paribus, computerization helps increase the concentration of selected consumer industries, such as retail chains, restaurants, and banking (Kling, 1978d). The net effect for consumers, would be not only dealing with the automated systems used by specific firms, but also having a narrower choice of standardized products from which to choose, and fewer firms to choose between. The opportunities for examining these interactions in relatively permeable service organizations such as chain stores, rather than courts and welfare agencies, will make investigation easier.

While these four sets of questions are not the only ones which might be asked about the development, use, or consequences of computing in organizations, they strike us as some of the more fruitful points of departure. This chapter has examined "computing as social action." Most attention to computing activities focuses on the equipment (hardware and software), and, secondarily, on the organizational procedures set up to manage the technology and the information flows it supports. Only the most meager attention is directed to the ways which people live and work with computing, and the ways they shape and are shaped by its uses. We have focused most of our attention to the latter, because we believe that it is the most neglected aspect of computing. People who work in complex organizations develop computer-based systems, embed them in their work, and find themselves and others dealing with the consequences. Computer technologies are not artifacts of nature like limestone caves. They are conceived, designed, shaped, ignored, tinkered with, layered, redesigned, sabotaged, criticized and appraised, to fit a complex web of human interests. That, in large part, is the interest and importance of computing.

Notes

1. This classification system is based on our reading of the literature on computing in organizations. Keen and Scott-Morton's (1978) book on decision-support systems examines the social dynamics of implementation from 5 different perspectives. Some of their categories differ from ours. They identify two sub-schools of the rational approach, "satisficing and "individual differences," which we cluster under the same common label. They ignore the structural, interactionist, and class politics perspectives. Labelling theoretical perspectives is more an art than a science. By spanning a wider array of theoretical perspectives than Keen and Scott-Morton, we have chosen to cluster some important variations of given perspectives under common labels.

2. These simulations utilize tables for material properties (e.g., specific heat), and special thermodynamic correlations for computing others. Properties of materials change in different tempature ranges. It is impossible to determine which values are used without careful inspection of the programs. This is not a trivial matter. Some of the chemical processes engineered by WESCO operate at cyrogenic tempatures while others operate near the boiling point of water. The chemical engineers select among a dozen thermodynamic correlations to compute other properties. These correlations are proprietary and an outside auditor cannot confirm whether the selected correlation for a particular analysis was appropriate. Even engineers within WESCO differ in their judgements about which correlations should be selected for a particular set of physical conditions. Processing plant design parameters vary as much as 20% depending upon the correlation chosen. The engineers typically work within 10% tolerances and selection of an appropriate correlation is technically and substantially important.

3. According to Kling and Scacchi (p. 108, 1979b) a computing package "includes not only hardware and software facilities, but also a diverse set of skills, organizational units to supply and maintain computing-based services and data, and sets of beliefs about what computing is good for and how it may be used efficaciously." Furthermore, an organization's computing package is situated within the shifting segments of the computing world and other relevant social worlds. A computing package is thus a processual ensemble of social and technical "fixtures" which encourage and constrain organizational computing activities (Kling and Gerson, 1978; Kling and Scacchi, 1979b; Scacchi, 1979).

4. OS 360 is not an information system, but is a large software system nonetheless.

5. Macro here refers to a programming construct.

6. The need for studies of errors in computerized transaction processing systems has recently emerged (Sterling, 1979). The relationship between mistakes in computing system use and errors encountered by the "end targets" of these systems is presently unexamined and poorly understood.

7. Simon waffles on the role of social settings in altering the roles played by computer-based systems. In <u>Organizations</u> (1958), he and March indicate that analytical problem-solving will occur under special organizational conditions. However, when he recommends the use of computer-based systems, he doesn't qualify his advice by attention to social context (Simon, 1977). See (Kling, 1978f) for a critique of Simon's analyses.

8. In addition to the studies described here, see Colton's (1979) studies of computer applications for police, the studies of Kling (1978b) and Laudon (1974) described earlier, and Stewart's (1971) studies of large-scale decision-support systems in British firms.

9. Sometimes closer supervision accompanies computing even though it is not accomplished through computing. One municipal water department installed an on-line inquiry system to help citizens learn about and negotiate their bills and payments. After the system was installed, the clerks who used it and who handled incoming phone calls had their telephone time monitored to ensure that they did not spend more than 2 minutes, on the average, in handling inquiries. Examples such as this lead casual observers to associate computer use with increases in supervision even when computing is not the instrument of enhanced supervision.

6.0 REFERENCES

Ackoff, R. (1967). Management Misinformation Systems Management Science 14 pp. 147-156.

Albrecht, G. (1976). The Effects of Computerized Information Systems on Juvenile Courts The Justice System Journal 2, pp.107-120.

Albrecht, G. (1979). Defusing Technical Change in Juvenile Courts Sociology of Work and Occupations 6, pp. 259-282.

Alter, S. and M. Ginzberg (1978). Managing Uncertainty in MIS Implementation, Sloan Management Review 19, (Fall, 1978), pp. 23-31.

Argyris, C. (1971). Management Information Systems:The Challenge to Rationality and Emotionality Management Science 17 pp. 275-292.

Argyris, C. (1973). Personality and Organization Theory Revisited Public Administrative Review 33, pp. 141-167.

Barkin, S. R. and G.W. Dickson (1977). An Investigation of Information System Utilization Information and Management 1, pp.35-45.

Belady, L.A. (1978). Large Software Systems, IBM RC 6966, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

Bell, D. (1973). The Comming of the Post-Industrial Society: A Venture in Social Forecasting, Basic Books, New York.

Bernard, D., J.C. Emery, R.L. Nolan, and R.H. Scott (1977). Charging for Computer Services: Principles and Guidelines, Petrocelli Books, New York.

Blau, P., C.M. Falbe, W. McKinley and P. Tracy (1976). Technology and Organization in Manufacturing, Administration Science Quarterly 21, pp. 20-40.

Boehm, B.W. (1973). Software and Its Impact: A Quantitative Assessment, Datamation 19, (May), pp. 48-59.

Boehm, B.W. (1976). Software Engineering, IEEE Trans. on Computers C-25, pp. 1226-1241.

Boehm, B.W. (1979). Software Engineering--As It Is, Proceedings Fourth International Conference on Software Engineering, pp. 11-21.

Braverman, H. (1974). Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century, Monthly Review Press, New York.

Brooks, F.P. (1975). The Mythical Man-Month, Addision-Wesley, Reading, MA.

Burch, J., F. Strater, and G. Grudnitski (1979). Information Systems: Theory and Practice Second ed. John Wiley, New York.

Cave, W.C. and A.B. Salisbury (1978). Controlling the Software Life Cycle -- The Project Management Task, IEEE Trans. Software Engineering SE-4, pp. 326-334.

Champine, G.A. (1978). Computer Technology Impact on Management, North-Holland, New York.

Colton, K. (ed.), (1978). Police Computer Technology Lexington Books, Lexington, MA.

Colton, K. (1979a). The Impact and Use of Computer Technology by the Police, Communications ACM 22, pp. 10-20.

Colton, K. (1979b). Police and Computer Technology - The Expectations and the Results, Proceedings 1979 National Computer Conference, Vol. 48, AFIPS Press, Montvale, NJ.

Cooper, J.D. (1978). Corporate Level Software Management, IEEE Trans. Software Engineering SE-4, pp. 319-326.

Cuff, E.C. and G.C.F. Payne (eds.). (1979). Perspectives in Sociology George Allen and Unwin, Ltd., London.

Danziger, J. (1977). Computer, Local Government and the Litany to EDP, Public Administration Review 37, pp. 28-37.

Danziger, J. (1979). The "Skill Bureaucracy" and Intraorganizational Control, Sociology of Work and Occupations 6, pp. 204-226.

Danziger, J. and W.H. Dutton (1977). Computers as an Innovation in American Local Governments, Communications ACM 20, pp. 945-956.

Danziger, J., W.H. Dutton, R. Kling, and K.L. Kraemer (forthcoming). Computers, Bureaucrats, and Politicians: High Technology in American Local Governments Columbia University Press, New York.

DeBrabander, B., D. Deschoolmeester, R. Leyder, and E. Vanlommel (1972). The Effects of Task Volume and Complexity on Computer Use Journal of Business 45, pp. 56-84.

DeMillo, R.A., R.J. Lipton and A.J. Perlis (1979). Social Processes and Proofs of Theorems and Programs, Communications ACM 22, pp. 271-280.

De Roze, B.C. and T.H. Nyman (1978). The Software Life Cycle -- A Management and Technological Challenge in the Department of

Defense, IEEE Trans. Software Engineering SE-4, pp. 309-318.

Dery, D. (1977). The Bureaucratic Organization of Information Technology: Computers, Information Systems, and Welfare Management Ph.D. Dissertation, Dept. of Public Policy, University of California, Berkeley, CA.

Downs, A. (1967). A Realistic Look at the Final Payoffs from Urban Data Systems, Public Administration Review 27, pp. 204-210.

Dutton, W.H. and K.L. Kraemer (1978). Management Utilization of Computers in American Local Governments, Communications ACM 21, pp. 206-218.

Dzida, W., S. Herda and W.D. Itzfeldt (1978). User-Perceived Quality of Interactive Systems, IEEE Trans. Software Engineering SE-4, pp. 270-276.

Edstrom, A. (1977). User Influence and the Success of MIS Projects: A Contingency Approach, Human Relations 30, pp. 589-607.

Galbraith, Jay (1977). Organization Design Addison-Wesley, Philipines.

Gerson, E.M. (1976). On "Quality of Life," American Sociological Review 41, pp. 793-806.

Gerson, E.M. and S.R. Koenig (1979). Information Systems Technology and Organizations: The Impact of Computing on the Division of Labor and Task Organization, working paper, Pragmatica Systems, San Fransisco, CA.

Gibson, C. (1975). A Methodology for Implementation Research in R. Schultze and D. Slevin (eds.), Implementing Operations Research/Management Science American Elsevier, New York.

Glass, R.L. (1977). The Universal Elixir and other Computing Projects which Failed, Computing Trends, Seattle, WA.

Goodman, P., J. Pennings, and Associates (1977). New Perspectives in Organizational Effectiveness Jossey-Bass, San Fransisco, CA.

Goodman, S. (1979). "Soviet Computing and Technology Transfer: An Overview" World Politics XXXI(4), pp. 539-570.

Gottlieb, C.C. and A. Borodin (1973). Social Issues in Computing, Academic Press, New York.

Greenbaum, J. (1976). Division of Labor in the Computer Field, Monthly Review 28, pp. 40-55.

Greenberger, M., M.A. Crenson and B.L. Crissey (1976). Models *in the Policy Process*, Russell Sage Foundation, New York.

Hedberg, B. and E. Mumford (1975). The Design of Computer Systems, in E. Mumford and H. Sackman (eds.) (1975). *Human Choice and Computers*, North-Holland, New York, pp. 31-59.

Herbert, S. and K. Colton, (1978). The Implementation of Computer-Assisted Police Resource Allocation Methods, in K. Colton (ed.), *Police Computer Technology* Lexington Books, Lexington, MA.

Hiltz, S.R. and M. Turoff (1978). *Network Nation: Human Communication Via Computer* Addison-Wesley, Reading, MA.

Jacob, F. (1977). Evolution and Tinkering, *Science* *196*(4295), pp. 1161-1166.

Kanter, J. (1977). *Management-oriented Information Systems* (Second edition), Prentice-Hall, Englewood Cliffs, NJ.

Keen, P.G.W. (1975). Computer-based Decision Aids: The Evaluation Problem, *Sloan Management Review* *16*, pp. 13-21.

Keen, P.G.W. and E. Gerson (1977). The Politics of Software System Design, *Datamation* *23*, (November), pp. 80-84.

Keen, P.G.W. and M. Scott Morton (1978). *Decision Support Systems: An Organizational Perspective*, Addison-Wesley, Reading, MA.

Kernigan, B.W. and J.R. Mashey (1979). The UNIX Programming Environment, *Software - Practice and Experience* *9*, pp. 1-15.

Kling, R. (1973). Toward a Person-Centered Computer Technology, *Proceedings 1973 ACM National Conf.*, Atlanta, GA.

Kling, R. (1974). Computers and Social Power, *Computers and Society* *5* (Summer), pp. 6-11.

Kling, R. (1977). The Organizational Context of User-Centered Software Design, *MIS Quarterly* *1*, (Winter), pp. 41-52.

Kling, R. (1978a). Information Systems and Policymaking: Computer Technology and Organizational Arrangements, *Telecommunications Policy* *2*, pp. 22-32.

Kling, R. (1978b). Automated Welfare Client-Tracking Service Integration: The Political Economy of Computing, *Communications ACM* *21*, pp. 484-492.

Kling, R. (1978c). EFT and the Quality of Life, *Proceedings 1978 National Computer Conference*, Vol. 47, AFIPS Press, Montvale, NJ, pp. 191-197.

Kling, R. (1978d). Value Conflicts and Social Choice in

Electronic Fund Transfer System Development, Communications ACM 21, pp. 642-656.

Kling, R. (1978e). The Impacts of Computing on the Work of Managers, Data Analysts, and Clerks, WP-78-64, Public Policy Research Organization, University of California, Irvine, CA.

Kling, R. (1978f). Information Systems as Social Resources in Policy-Making, Proceedings 1978 National ACM Conf., Washington, DC.

Kling, R. and E. Gerson (1977). The Social Dynamics of Technical Innovation in the Computing World, Symbolic Interaction 1, pp. 132-146.

Kling, R. and E. Gerson (1978). Patterns of Segmentation and Intesection in the Computing World, Symbolic Interaction 1, pp. 24-43.

Kling, R. and W. Scacchi (1978). Assumptions about the Social and Technical Character of Production Programming Environments, Proc. Irvine Workshop on the Alternatives for the Environment, Certification, and Control of the DOD Common High Order Language, Dept. of Information and Computer Science, University of California, Irvine, CA.

Kling, R. and W. Scacchi (1979a). The DoD Common High Order Programming Language Effort (DoD-1): What Will the Impacts Be? SIGPLAN Notices 14, (February), pp. 29-43.

Kling, R. and W. Scacchi (1979b). Recurrent Dilemmas of Computer Use in Complex Organizations, Proceedings 1979 National Computer Conf., Vol. 48, AFIPS Press, Montvale, NJ, pp. 107-116.

Kling, R., W. Scacchi and P. Crabtree (1978). The Social Dynamics of Instrumental Computer Use, SIGSOC Bulletin 10, (Summer), pp. 9-21.

Kraemer, K.L. (1977). Local Government, Information Systems, and Technology Transfer, Public Administration Review 37, pp. 368-382.

Kraemer, K.L. and W.H. Dutton (1977). Technology and Urban Management: The Power Payoffs of Computing, Administration and Society 9, pp. 305-340.

Kraemer, K.L. and W.H. Dutton, (1979). The Interests Served by Technological Reform: The Case of Computing, Administration and Society 11, pp. 80-106.

Kraemer, K.L., W.H. Dutton, and A. Northrop (forthcoming). The Management of Information Systems, Columbia University Press, New York.

Kraemer, K.L. and J.L. King (1979). A Requiem for USAC, *Policy Analysis 5* pp.313-349.

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

Laudon, K. (1974). *Computers and Bureaucratic Reform*, Wiley-Interscience, New York.

Lawler. E.E. and J.G. Rhode (1976). *Information and Control in Organizations* Goodyear Pub., Pacific Palisades, CA.

Learmonth, G.P. and A.G. Merten (1978). Operation and Evolution of Organizational Information Systems, in G. Bracchi and P.C. Lockeman (eds.), *Information Systems Methodology*, Springer-Verlag, New York, pp. 664-683.

Licklider, J.C.R. and A. Vezza (1978). Applications of Information Networks, *Proceedings IEEE 66*, pp. 1330-1346.

Lientz, B.P., E.B. Swanson, and G.E. Tompkins (1978). Characteristics of Applications Software Maintenance, *Communications ACM 21*, pp. 466-471. pp. 517-524, (1978)

Lucas, H. (1974). *Toward Creative System Design*, Columbia University Press, New York.

Lucas, H. (1975). *Why Information Systems Fail*, Columbia University Press, New York.

Lucas, H. (1978). Unsuccessful Implementation: The Case of a Computer-based Order Entry System, *Decision Sciences 9*, pp. 68-79. pp. 197-205.

Mann, F.C. and L.K. Williams (1960). Observations on the Dynamics of Change to Electronic Data Processing Equipment, *Adminstrative Science Quarterly 5*, pp. 217-256.

Markus, M.L. (1979). *Understanding Information System Use in Organizations: A Theoretical Explanation*, Ph.D. Dissertation, Department of Oragnizational Behavior, Case Western Reserve University, Cleveland, OH.

Mashey, J.R. and D.W. Smith (1976). Documentation Tools and Techniques, *Proceedings Second Inter. Conf. Software Engineering*, pp. 177-181.

Maynard, J. (1979). A User-Driven Approach to Better User Manuals, *Computer 12*, pp. 72-75.

McLaughlin, R.A. (1978). The (mis)use of DP in Government Agencies, *Datamation 24*, (July), pp. 147-157.

Mills, H. (1976). Software Development, *IEEE Trans. Software*

Engineering SE-4, pp. 265-273.

Mills, H. (1977). Software Engineering, Science 195, pp. 1195-1205.

Mowshowitz, A. (1976). The Conquest of Will: Information Processing in Human Affairs, Addison-Wesley, Reading, MA.

Mumford, E. (1972). Job Satisfaction: A Study of Computer Specialists, Longman, London.

Mumford, E. and O. Banks (1967). The Computer and the Clerk, Routledge and Kegan Paul, London.

Mumford, E. and A.M. Pettigrew (1976). Implementing Strategic Decisions, Longman, New York.

Mumford, E. and H. Sackman (eds.) (1975). Human Choice and Computers North-Holland, New York.

Myers, C. (1970). Computers in Knowledge-Based Fields, MIT Press, Cambridge, MA.

Noble, D. (1978). "Social Choice in Machine Design: The Case of Automatically Controlled Machine Tools, and a CHallenge for Labor. Politics and Society 8(3-4):313-47.

Oliver, P (1978). Guidelines to Software Conversion, Proceedings 1978 National Computer Conference, Vol. 47, AFIPS Press, Montvale, NJ, pp. 877-886.

Palme, J. (1978). How I Fought with Hardware and Software and Succeeded, Software - Practice and Experience 8, pp. 77-83.

Perrow, C. (1979). Complex Organizations: A Critical Essay (2nd ed.) Scott, Foreman, and Co., Glenview, IL.

Perry, J.L. and K.L. Kraemer (1978). Innovation Attributes, Policy Intervention, and the Diffusion of Computer Applications among Local Governments, Policy Sciences 9, pp. 179-205.

Pettigrew, A.M. (1972). Information Control as a Power Resource, Sociology 6, pp. 187-204.

Pettigrew, A.M. (1973). The Politics of Organizational Decision-Making, Tavistock, London.

Riemer, J.W. (1976). "Mistakes at Work" The Social Organization of Error in Building Construction Work, Social Problems 23, pp. 255-267.

Robey, D. (1979). MIS Effects on Managers Task Scope and Satisfaction, Proceedings 1979 National Computer Conference, Vol. 48, AFIPS Press, Montvale, NJ, pp.391-395.

Rule, J. (1974). Private Lives and Public Surveillance: Social Control in the Computer Age, Schocken Books, New York.

Rustin, R. (ed.) (1972). Debugging Techniques in Large Systems, Prentice-Hall, Englewood Cliffs, NJ.

Sandewall, E. (1978). Programming in an Interactive Environment: the "LISP" Experience, Computing Surveys 10, pp. 35-71.

Scacchi, W. (1980). The Process of Innovation in Computing: A Study of the Social Dynamics of Computing, Ph.D. Dissertation, Dept. of Information and Computer Science, University of California, Irvine, CA.

Sheiderman, B. (1978). Improving the Human Factors Aspect of Database Interactions, ACM Trans. Database Systems 3, pp. 417-439.

Simon, H.A. (1947). Administrative Behavior, Macmillan and Co., New York.

Simon, H.A. (1965). The Shape of Automation for Men and Management, Harper and Rowe, New York.

Simon, H.A. (1973). Applying Information Technology to Organizational Design, Public Administrative Review 33, pp. 268-278.

Simon, H.A. (1977). The New Science of Management Decision Prentice Hall, Engelwood Cliffs, NJ.

Sterling, T. (1979). Consumer Difficulties with Computerized Transactions: An Empirical Analysis, Communication ACM 22, pp. 283-289.

Stewart, R. (1971). How Computers Affect Management, MIT Press, Cambridge, MA.

Strauss, A. (1978a). Negotiation, Josey-Bass Inc., San Francisco, CA.

Strauss, A. (1978b). A Social World Perspective, in N. Denzin (ed.), Studies in Symbolic Interaction, Vol. 1, JAI Press, Chicago, IL, pp. 99-128.

Strassman, P.A. (1973). Managing the Evolution to Advanced Information Systems, in F.W. McFarland, R.L. Nolan, and D.P. Norton, Information Systems Administration, Holt, Rhinehart, and Winston, New York.

Streeter, D. (1975). The Scientific Process and the Computer Wiley Interscience, New York.

Walston, C.E. and C.P. Felix (1977). A Method for Programming

Measurement and Estimation, IBM Systems Journal 16, pp. 54-73.

Whisler, T.L. (1970). The Impact of Computers on Organizations, Praeger, New York.

Wolverton, R.W. (1974). The Cost of Developing Large-Scale Software, IEEE Trans. on Computers C-23, pp. 615-636.

Youngs, E.A. (1974). Human Errors in Programming, Inter. Jour. Man-Machine Studies 6, pp. 361-376.