Behind the Terminal: The Critical Role of Computing Infrastructure In Effective Information Systems' Development and Use

Kling, Rob

Behind the Terminal:
The Critical Role of Computing Infrastructure
In Effective Information Systems' Development and Use

Rob Kling

Center for Research on
Information Technology and Organizations
(CRITO)
and
Department of Information
and Computer Science

ABSTRACT

Contemporary approaches to systems analysis ignore the importance of computing infrastructure -- the kinds of resources necessary for making computerized system workable and effective. Infrastructure includes "hard resources" such as electricity and physical space; it also includes human resources such as the skill levels of systems users and maintainer.

Systems analyses which account for infrastructure can help lead to more effective recommendations. The key organizing ideas of this paper, web models, are based on almost 20 years of empirical studies of the ways that people and organizations adopt, develop and use computerized systems. It is based on an understanding of how people and organizations actually behave rather than upon a model which prescribes how they should behave.

Web models draw "large" social boundaries around a focal computing resource so that the defining situation includes: the *ecology of participants* who influence the adoption and use of computer-based technologies, the *infrastructures* for supporting system development and use, and the *history* of local computing developments. Social action characteriwed by "natural open systems" models of organizations.

Web models help explain the actual leverage of computing developments, their carrying costs, and the ways that systems are valued by different participants. Web models are contrasted with conventionally rational "discrete-entity" models which are a-contextual, a-historical, and assume that adequate infrastructure can always be available as needed.

The web approach to computing infrastructure is illustrated with a rich longitudinal case study of the use of desktop computing in a dynamic work group. The chapter also examines ways that infrastructure designs influence the choice of computerized systems which are appropriate for specific groups of users.

Acknowledgements: I've deepened my understanding of computing infrastructure and systems development in discussions with Jon Allen, Werner Beuschel, Joey George, Suzanne Iacono, Tom Jewett, Matthew Jones, Kate Kaiser, John King, Kenneth Kraemer, Steve Lepore, Geoff Walsham, and Mary Zmuidzinas. Funds for conducting some of the research reported here came from NSF Grants IRI8709613 and IRI9015497.

1.0 INTRODUCTION

In this paper I discuss the development and use of computerized systems in organizations in ways that can help systems analysts make more effective recommendations. The key organizing ideas of this paper are based on almost 20 years of empirical studies of the ways that people and organizations adopt, develop and use computerized systems. It is based on an understanding of how people and organizations actually behave rather than upon an abstract model of how they should behave (Kling and Scacchi, 1982; Kling, 1987; Dunlop and Kling, 1991).

Computerizing an organizational activity involves much more than mapping information flows, finding equipment to automate work and decisions, installing the equipment, and training users. Many computerization projects lead to new work practices and new divisions of labor. They may raise (or at least alter) standards for acceptable goods and services. They may lead people to expect themselves and their coworkers to work differently and often faster. Many analysts assume that the "technologically best equipment" will lead to the most significant improvements, even though professionals may differ over which features of work and organizations should be viewed as worst and best. The "surface features" of computerization are the most visible and the primary subject of debates and systems analysis. But they are only one part of computerization projects. Many key parts of information systems are neither immediately visible or interesting in their novelty.

An analogy with urban architecture is illustrative. We often think of the architecture of cities like New York in terms of skylines and the design of major buildings. For me, the phrase "New York City architecture" evokes images of the imperial skyscrapers like the Empire State Building, the World Trade Center, the AT&T building, and the Guggenheim Museum, as well as residential streets with brownstones. While the variety of business life in New York City depends upon skyscrapers, it also depends upon numerous services that are invisible from the street including electric lines, telephone cables, and sewer lines. It depends upon "transportation systems" including cabs, buses, and subways. This *urban infrastructure* facilitates business life in the office buildings. Without electricity and communication lines, business in the most imperial of skyscrapers would grind to a halt as elevators and computers stopped, and the only way to contact someone would be by walking up and down 10, 20 or even 50 flights of stairs. If the cabbies, subway engineers, and bus drivers all quit work simultaneously, the streets would be clogged with cars. Despite the city's dependence on a complex and refined urban infrastructure, much of it is out of sight, taken for granted, treated as dirty, and not a primary part of our image of "business support in New York City."

Another analogy comes from theater and film. We normally focus on the visible elements of a plays and films: the cast, director, writers (script) and sets. The actual production of a play or film involves numerous other people to design costumes, coach actors, locate sets, design lighting, and so on. In the 1980s, films began to give credit to a much larger group of these essential "backstage participants." One can now see film credits for hair designers, voice coaches, and grips, as well as the traditional cast, writers, and directors.

Workable computing arrangements also depend upon a set of supporting resources, which can be physical, technological or social. Physical resources include the space to place equipment; technological resources includes electricity and communication lines. The social resources include people skilled in using and repairing equipment and practices for allocating resources. I call this collection of resources **computing infrastructure** .

Computing infrastructure is a key element in any computerization effort (Kling and Scacchi, 1982; Gasser, 1986; King and Kraemer, 1986; Kling, 1987; Clement, 1990; Clement and Parsons, 1990; Jewett and Kling, 1990; George, Iacono, and Kling, 1991; Jewett and Kling, 1991). People report having better quality worklives in workgroups where the computing infrastructure for supporting training and consulting is best developed (Lepore, Kling, Iacono, and George, 1989). Further, poorer organizations, like many public schools, sometimes have trouble effectively using gifts of advanced computer systems because of weakness in their computing infrastructure. Training is one aspect of computing infrastructure which receives some attention. But computing infrastructure refers to a much richer array of resources and practices. This chapter examines the role of infrastructure in computer use and suggests how systems analysts might include an analysis of computing infrastructure in their recommendations about appropriate computing arrangements.

Today, there are two dominant approaches to systems analysis:

1. Formal information processing approaches which emphasize the information processing capabilities of computer systems as instruments for manipulating data or making decisions. These approaches (such as those which map information flow with charts and flow diagrams) focus on formal tasks and generally ignore the social relationships between people and organizations who develop, maintain and use computerized systems. They also assume that people should follow certain narrow rationalistic norms, and tacitly assume that people and organizations will devote whatever resources are necessary to make computing systems effective. These approaches can be useful under special conditions, but they do not help analysts adequately understand the conditions under which people are likely to use systems and the ways that likely patterns of use should influence the *social designs* of information systems. Further, these approaches focus on the most visible aspects of information systems and ignore the resources required to build and operate them effectively.

2. Social process approaches, such as socio technical systems, Christiansen's (1990) activity approach, Checkland's soft systems (Checkland, 1992), help people and organizations better identify their preferences (so called *requirements*). Process analysts focus on ways of defining problems involving a wide variety of participants in the conception and design of systems. Their strength is to build commitment from people who are likely users of computerized systems, to better identify participant's preferences, and to help developers and users learn about each other's work and the systems which facilitate them. However, they rarely bring key organizing concepts about computer systems, such as infrastructure, to these design meetings.

In this chapter I advocate the use of a specific set of ideas about computing infrastructure in organizations which could help expend the intellectual repertoire of rationalistic designers and enrich the design discussions of social processes designers. Any systems analysis must characterize the computer based technologies, the social settings where they are used, and the social forces which shape their use. Systems analyses differ in four ways which have a tremendous influence on their recommendations:

1. the scope of the boundaries they draw around a computer based system and the key actors identified developers, users, regulators, maintainers, promoters, system saboteurs, clients of computer using work groups, etc.;
2. the extent to which they examine social relations between key participants, such as the kinds of interdependence, control, cooperation and conflict;
3. the extent to which they examine resources that key participants have to carry out important lines of action; and
4. the extent to which they examine actual social meanings which participants bring to their encounters with computer-based technologies.

Unfortunately, systems analyses for computing rarely have explicit characterizations of technologies other than information processing models. This chapter examines how a sociological characterization of the computing infrastructure and organizations which develop, adopt, and use them can improve the quality of systems analysis.

This chapter examines two explicit models which underlie much of the thinking about computer systems in organizations. These models help identify some of the key choices that systems analysts often make along the four dimensions identified above. One class of models (mentioned above) focuses on relatively formal-rational conceptions of the capabilities of information technologies and the social settings in which they are developed and used. These conceptions, which we label *discrete-entity* models, focus on explicit economic, physical, or information processing features of the technology. The social context in which the technology is developed and used, and the history of the participating organizations, are limited to a few formal relationships or are ignored. Systems are assumed to be loosely aggregated collections of equipment, people, procedures, and beliefs, which may easily be broken down into separate elements, and costed and evaluated independently. Social relations are assumed to be cooperative and resources are available as needed. Discrete-entity analysts also assume that "good" social actors will value "better" technologies. These conceptions are most common in professional and academic publications which emphasize positive management.

The second class of models, *web models*, are a form of naturalistic, open systems models (Scott, 1987a). They make explicit connections between a focal technology, its immediate users, and a rich ecology of social relationships with other social groups and organizations in which the technology is developed, adopted and used (Kling and Scacchi, 1982; Kling, 1987; Wood-Harper and Corder, 1988). Computer systems, in this conception, are developed, operated, and used by an interdependent network of producers and consumers and cannot be adequately analyzed solely according to their discrete features and components. Web models treat computerized systems as a form of social organization with important information processing, social, and institutional properties: they are not only flexible information processing tools. Their "shape", the way they are used, the leverage they provide, and the interests they serve depend upon the interplay of stakeholders, resources, and social games within which they are deployed. Their capabilities and fit within organizations depend upon these social relationships as well as upon their information processing characteristics.

Web models of computing emphasize the history of commitments made in the course of computing deployments and the infrastructure that supports the deployments of computer-based technologies. They draw "larger boundaries" around the focal computing resources than do discrete entity models, by examining how their development and use depend upon a social *context* of complex social actions in which the focal computing resources is developed or used. Web models define a social context for a computer-based technologies by taking into account:

> the *social relations between a set of participants* who can influence the adoption, development, or use of the focal computing technologies;
> the *infrastructure* available for their support;
> the *history* of commitments made in developing and operating related computer-based technologies.

These three elements will serve as key organizing concepts for web analyses. In addition, they point to useful methodologies for doing web analyses. We discuss appropriate methods briefly near the end of the chapter.

Discrete-entity models and web models represent ideal constructs. By making them explicit and sharpening their differences we can contrast the analytic power of each model and ask about the insights each model provides into the sagas of computing in organizational life. Discrete entity models have dominated systems analysis to date. We argue that web models better account for critical aspects of computing developments: their costs and effectiveness, their speed of planned change, and their integration into organizational life. In the following sections we describe these two models in more detail, report a case study of computerization which helps illustrate key concepts, and examine key aspects of computing infrastructure with examples from the case.

2.0 MODELS OF COMPUTERIZATION: DISCRETE-ENTITY AND WEB

In contrasting discrete-entity and web models, it is first useful to examine their underlying assumptions in more detail, and to identify their implications. Table #1 lists five key assumptions of discrete entity models.

==

Table #1 About Here

==

The basic units of analysis of discrete-entity models are computing resources (e.g., an Apple MacIntosh computer, a specific computer program) and the formal tasks to which they are applied. Analysts who employ discrete-entity models assume instrumental rationality, and base their predictions and diagnoses on two logics of technical development: piecemeal aggregation and direct substitution. A simple example of these attractive assumptions can be illustrated when participants describe their computing environments by a listing of equipment.

Suppose that the staff of a university department describe their computing environment by listing this set of equipment:

Microcomputers: 40 diverse IBM PC's, 5 diverse Apple MacIntoshes;
Word processors: Microsoft Word 5, WordPerfect 5.1.;
Dot matrix printers: 1 Apple Imagewriter, 20 Epson FX86 (9 pin);
Laser printers: 2 Apple LaserWriters;
Databases: Paradox 3.5 and Dbase-IV; *and*
Spreadsheets: Microsoft Excel

The first logic of technical development, piecemeal aggregation, assumes that new components add new capabilities whose *best features are available for all participants*. For example, by aggregation, one could assume that any microcomputer user in this department could use a laser printer. In practice, diverse sets of equipment like those on this list are usually organized into several collections that work well together. But the ensembles may not work well together. For example, suppose that the faculty use the IBM PC's and have the dot matrix printers. Suppose that the departmental secretaries and office staff use the Macs and share the laser printers. The Apple laser printer may work well with Microsoft Word on the Macs, but not with applications run on IBM PC's. Documents which the office staff prepare may routinely appear around the university with distinctive laser printed fonts. University administrators might assume that the faculty also can use the LaserWriter. But some of the PC software may not have drivers for the LaserWriters varied fonts. On the other hand, the faculty may be able to have staff support with budgeting on spreadsheets since they all use Microsoft's Excel. This department may look computationally richer than one which has no laser printers. But the factoring of the milieu into somewhat distinct computing worlds means that only one subworld is "rich" in laser printers. Conversely, only the users of the IBM-PC's may be able to use the databases.

The second logic, direct substitution, assumes that the gains realized by replacing one computing device by a more powerful are the same for all participants in all settings from a testing laboratory to an office. For example, suppose that the department chair considers two proposals:

a. Replace three of the 9 pin Epson dot matrix printers with an equal number of 24 pin Epson dot matrix printers;
b. Replace the slowest IBM PC's with the fastest IBM PC clones available.

These two proposals illustrate the archetypical case of direct substitution; a better component replaces a lesser quality component and users of the components can exploit the information processing capabilities that the new component offers. A third proposal, to replace two Epson dot matrix printers with two Apple LaserWriter, may also appear as direct substitution. At one level of abstraction this substitution on the department's computer inventory list may make it appear to be computer richer. However, in terms of the day to day work of faculty and staff, this third proposal is redistributive. Some faculty would lose printing capabilities because they would relinquish their dot-matrix printers and have to share the remaining printers while the departmental staff would gain access to additional laser printers. A fourth proposal, to replace the IBM PC's with more powerful engineering workstations based which run Unix System V would also appear as enriching on paper. But if the faculty who used the PC's lost access to important software (like Paradox 3.5 and a strong word processor) in the shift from the PC's to the workstations, they could end up computationally poorer (e.g., "hardware rich and software poor").

These technological illustrations suggest the difficulties of examining the value of computing resources as isolated entities, rather than as parts of larger systems and specific organizational arrangements. In the case of this hypothetical university department, we have examined the interplay of technical features. However, we should examine the interplay of both technical and social features a key feature of web models.

==

Table #2 About Here

==

Five critical assumptions of a web model are listed in Table 2. Web analysts are concerned with the array of activities that people actually engage in while pursuing some task, rather than examining isolated components of a computer based system and assuming that they will be smoothly grafted into a formal task system. Thus, a web analyst would not start by asking "what kind of equipment is being used?" or "what are the formal goals of this organization?," but "who are the key actors, what kinds of things do they do here, what incentives influence their activities, and what organizational routines constrain their actions and choices?" The web analyst identifies the local ecology of games (Long, 1958) and local behavioral constraints to examine how they influence computerization. Assumptions W2 and W3 bear directly on the complexity of social arrangements into which computing is embedded and the extent to which different participants control key resources. These analyses are *explicit* rather than implicit.

The logical framework of web analyses rests on the assumption that a computing resource is being used by an interdependent network of producers and consumers. We call this network a production lattice. The production lattice for a particular computer-based system or service is a social organization which is embedded in a larger matrix of social and economic relations and is dependent on a local infrastructure. According to web models, broader social relations and the local infrastructure shape the kind of computer-based service made available at each node of the production lattice. These "broader social relations" specially include 1) the inter-dependencies between different groups who develop and use a computing resource and 2) other social agents who most depend upon the computing resource or upon whom the resources' users most depend.

Workable computing arrangements depend upon the infrastructural resources available. Some of these infrastructural resources, such as equipment and staff with specialized technical skills are difficult to replace instantly. They evolve over time. Computing developments are shaped by a set of local commitments to technologies and social arrangements to support them which develop over time. In short, web models view computer-based systems as complex social objects whose architecture and use are shaped by the social relations between influential participants, the infrastructure that supports them, and the history of commitments. Web analyses focus explicitly on each of these influences.

This chapter examines the relationship between computing infrastructure, computer use, and organizational behavior based on web models. Analysts who use discrete-entity models can argue that computer-based systems are powerful media - for good or bad. Analysts who use a discrete-entity model can emphasize *new capabilities* and *new opportunities* by focusing on the capabilities of computing resources while neglecting the available infrastructure, the context of likely use, or the history of computing commitments which may constrain a computer-using organization. Social boundaries that are drawn tightly around a computer system and a single user can make it appear that the user can gain all the information processing benefits that the system can provide. A boundary drawn tightly around the administrative domain of a manager can make it appear that he is in control of his organizational unit's destiny[1].

Focusing on the context of computer use, the infrastructure which supports computer use, and the history of computing developments usually adds constraints and conditions which weaken the direct links between computer use and some social outcome whether it is socially desirable or troublesome. If new computing resources or organizations which adopt them depend upon the resources or cooperation of others, then it less certain that participants will experience the most extreme capabilities of computing. Examining the available resources further (e.g., assumption W3a) can also mute enthusiasm or criticism. Moreover, assumption W4b focuses on social practices which can highlight conflict, as well as cooperation. Assumption W4b also points to activities which can stigmatize organizations if they are evaluated in a conventional moral perspective. Overall, these foci make web analyses less intensely exciting or frightening than the hopes of liberation or fears of dehumanization which are often married to new computing developments in discrete-entity analyses.

3.0 THE CASE OF THE DESKTOP COMPUTING RESEARCH PROJECT (DCRP)

Concrete examples can help illustrate key ideas of web analysis. The following case of computerization to support the work of a research team illustrates key ideas about the management of infrastructure. Infrastuctural resources can be developed or "accepted" from the web of social relations within which a group works. The nature of available infrastucture should influence key choices about ways to computerize, but it can also requires continual attention for effective computer use. This case differs from many traditional information systems cases since the research team does not use one major information system. Rather, they develop and use many small information systems, including their collections of survey and interview data. And they use computing for some of their primary tasks, such as writing memos and papers and communicating via electronic mail. They represent a major new style of computerization in professional work groups with multiple systems used for diverse activities. The details of this work group differ from those of many other organizations, in terms of their specific activities, resources, technologies, and social relationships. But the case helps illustrate *key processes* which are relevant to many computerization projects: the ways that groups can depend upon others for key resources, the ways that affordable computing infrastructure helps shape the computing systems adopted, the ways that resources are negotiated, and the ways that important commitments become institutionalized.

The Desktop Computing Research Project (DCRP) is a research team whose subject of inquiry is the ways that computerization alters work. The DCRP team uses computing for a variety of tasks, including writing memos and papers, keeping research records, analyzing extensive survey data from nearly 1000 respondents, and communicating via electronic mail. Even though I am the faculty principal investigator (PI) of the DCRP, I am reporting the case in the third person to help create distance between the analysis of infrastructure and my narrative of this case. Further, this case is based on substantial observations by other project members and is not anchored only in my own observations[2].

The PI held a research professorship at PPRO, in addition to his primary academic appointment in ICS, since 1973. He had been principal investigator on other research projects which were administered through PPRO. The PI initiated the DCRP in 1985 as a longitudinal study of computerization and changes in work life. The PI holds his primary faculty appointment in the Department of Information and Computer Science (ICS). The project is administered by a campus social science research institute --Public Policy Research Organization (PPRO). PPRO served as a home for about 30 faculty-lead research projects, in diverse fields, including social aspects of information technology, environmental planning, and community mental health. PPRO provides its research projects with office space, and recharges for grant management, statistical computing and secretarial work. PPRO provided **much stronger** grants management and liaisons with Federal funders than did ICS, even though its office services were of comparable quality. The PI valued PPRO's grants management and was willing to manage the resulting complexities of working in two organizations which were in different building complexes and which relied upon incompatible computing environments.

The project team varied in size between 3 and 8 people during the six years of its duration between 1985 and 1991. It consists of the PI, between one and three Research Associates and Assistants (Ras) with bachelors or masters degrees, several undergraduate assistants (paid and unpaid), and a part-time office assistant. One important social characteristic in university-based research teams is the wide variation in turnover rates. The PI developed the DCRP along with one full time research associate who helped define key research methods during her three and a half years on the team. Graduate RAs work with the DCRP for two or three years. The paid undergraduate staff may work for up to a year, seldom longer; many have worked for just one or two ten-week terms. The independent-study students come and go term-by-term; a few have remained for two or three terms. The office assistants usually work on the project for one academic year, although one assistant stayed for two and a half years.

One key infrastructural feature of this research team is the computing expertise of the individual members. The PI is a computer scientist; the RAs have come from both computer science and social science. More important, the undergraduates and even some graduates accustomed to an individualistic student life vary widely in their abilities to coordinate effectively and work reliably in the DCRP. The overall computing expertise within the DCRP is at least equal to that of most research (or business) work groups. But expertise is not evenly distributed among work group members. One staff member is an expert in computer operating systems and utility programs. Another is an experienced applications programmer. Yet another is proficient in operating and interpreting the DCRP's statistical package. Some of the undergraduates have had specialized hardware maintenance experience. But the DCRP does not have the luxury of specialization: the DCRP's members use most of the applications software to accomplish their tasks. Some staff have had to learn virtually all of the project's software after joining. Even a group with significant computing expertise can experience many subtle complexities and anomalies which add unexpected time and skill demands to the continuous process of computerization.

3.1 Tasks and Resources

The DCRP's computing equipment and management strategies were chosen to facilitate its specific mix of tasks with equipment that the PI could afford in 1985 and which would enable project staff to work easily at home and their campus offices. This computing equipment was upgraded during the course of the project, but compatibility with earlier equipment was a major consideration in systems changes.

3.1.1 Data Complexity. The central tasks of the DCRP were to study computerization and changing work by conducting surveys and ethnographic studies and publishing the results in conferences, journals, and book chapters. The study was designed by the PI and a senior research associate. During the study, one or two research assistants helped collect data through interviews, distributed and collected surveys, and analyzed survey data.

During the study, the DCRP staff collected a significant amount of interview and survey data, which was facilitated by the availability of computing, and which also influenced key patterns of computer use. The complexity of a research computing environment can be driven by the data required to conduct a specific project. The DCRP's data environment has become increasingly complex as the multi-year project has progressed. Between 1985 and 1991 the DCRP collected a substantial body of research data (interviews, surveys, and article abstracts) and administrative data (site information, professional mailing lists).

For example, the first wave of survey data for the longitudinal survey was stored in a 600KB file containing the responses of 300 individuals, to a questionnaire with about 200 items. and a second file which included summary indices of key measures of computerization and work life. In the second and third years (1989 and 1990) the DCRP added similar data sets, for a total of six master files and numerous derivative files to be maintained, compared, and analyzed for changes over time. The site contact data base grew during the project to accommodate changing organizations and contacts. It contains information about 12 organizations, over 40 work groups, over 60 key individuals, and survey details. It is organized in a database of 12 linked files -- each contains from 10 to 20 separate data fields and from 50 to 200 individual records. Other DCRP documents -- papers, memos, reports, and correspondence -- multiplied significantly. The complexity of the DCRP's data environment is reflected in the project's computing hardware and software.

3.1.2 Money. The DCRP was funded primarily by grants from a federal agency (National Science Foundation) and received most of its computing equipment from the PI's academic department, ICS. The grants provide funds for personnel, supplies, administration, travel, services, computing software and computer operations that are needed to conduct research and to disseminate research results. The PI obtained most of the DCRP's 8 PCs through funding from his academic department. However, he competes with other departmental research groups for allocation of hardware (through a committee process), and has negligible control over the mix of resources procured by the department or the schedule on which they become available. The DCRP benefitted from the technological tastes of ICS faculty. Most of them value engineering workstations or Apple MacIntosh computers, and the DCRP had little competition in seeking 80286-based PCs which the ICS department allocated to faculty and staff in 1988 and 1989. (The DCRP had some significant compeition in obtaining 80386-based computers from ICS in 1989-90 because other faculty believed that they could use them as file servers). But the DCRP has little direct computing support from ICS, since their computer support staff focus on numerous networked computers which run Berkeley Unix and (more recently) several instructional Mac labs. The campus computing facility provides statistical computing on a DEC VAX minicomputer, but it's use is relatively expensive.

The PPRO office can provide some computing services (e.g., statistics with SPSS on a dedicated IBM minicomputer and word processing with Microsoft Word on PCs) but not others (such as database support for field contacts and abstracts). These services would be recharged to the DCRP, and would also require specific computer file formats to match PPRO's systems. Finally, the PI and some RAs privately purchased PCs for home use that they utilize for some research tasks. The DCRP's control over resource decisions -- especially computing resources -- depends on the funding source (NSF, ICS, PPRO, or private purchase), the total budget allocated to computing, and the actual dollar amounts involved. Like many groups, the DCRP has much greater discretion in smaller purchases (e.g., a $200 software package) than in major ones (such as hardware costing over $1,000).

3.1.3 Physical Environment. The DCRP's office space was stable, even though the size of the DCRP's research team has varied. The DCRP is administered by a social science research institute --Public Policy Research Organization (PPRO). PPRO provides its research projects with office space, and recharges grants for administration, statistical computing and secretarial work. The DCRP staff shares a large partitioned office with other research groups. The DCRP's own small (160 sq. ft.) office is delineated by modular, movable partitions and is configured for 5 or 6 workers, each of whom has a small desk, a chair, some limited file and shelf space, and a PC. Some assets are devoted to common use: one table, several file cabinets, and additional shared computing equipment for specialized applications. In contrast, major businesses often provide private space for workers at the level of the PI and graduate RAs — perhaps over 700 square feet for the present team composition. In addition, the project staff used one shared phone line with several extensions.

The cramped space which PPRO allocates for the DCRP's office has largely determined many other structural conditions of work life and choice of computing arrangements. The RAs find that working in the office can be distracting -- especially when others are in. Crowding encourages discussion, but reduces the opportunities for private thought, data analysis, or writing. As a result, some of the RAs do much of their analytical work and writing on PCs at their homes. The PI has a departmental office (in another building), but no private space in the research area. Although he could use some of the DCRP's desk space, he has ceded this to the other project staff and often works at home. In addition, he set up an additional computer in a second home office for a part-time office assistant. Given the small research office and the staff's unpredictable academic schedules, coordination among team members requires flexibility and ingenuity. The PI and RA's use electronic mail daily, frequently discuss issues via telephone; frequently share files through a Sequent mainframe computer; and meet at least once a week in a project-wide staff meeting. Managing this environment is an important element of the DCRP's computing strategy.

3.2 Computerization Strategies

Computerization strategies in a work group are more complex than simply selecting system components. They include practices of controlling access to equipment and data and infrastructure development (e.g., training, equipment repair) (Kling and Iacono, 1989b). The preferences of research groups as they develop or enhance their computing environments may be incompatible or even mutually exclusive. The PI and key RAs of the DCRP organized computing in accord with these five preferences:

3.2.1 *Cost containment:* Many research managers work within fixed budgets and cannot recharge their funders when they do more work. PIs resolve competing demands for their fixed budgets between items such as staff salaries, supplies, contracted services, and travel for research or conferences. Computing resources are usually subordinate. Two types of computing expenses must be balanced: capital expenditures (i.e., equipment procurement) and operating expenditures (for computing services, supplies, staff, etc.). Funding for capital and operating expenses may each come from different sources; each source has different restrictions about how much money can be spent for different categories of expenses. The DCRP's operating budget has been a key constraint. The DCRP has found in-group computing to be more consistent with its budget than outside sources -- especially in document preparation and statistical analysis.

The PI also valued knowing the range of likely expenditures, as well as their absolute magnitude of expenses. He found it difficult to get accurate cost estimates for a variety of tasks from the local service providers. Further, in a recharge system with delayed reports, some costs can mount unexpectedly. For example, student assistants can spend a large fraction of the annual computing budget during a few fiendish weeks of statistical analysis, and the bills may not show up for well over a month. Since the PI wanted to encourage students to experiment with data analysis and not have to exercise tight control over detailed analyses, he preferred statistical facilities which had low marginal costs for additional use. As a result, the DCRP has continued to control its own computing applications at the cost of some staff time and infrastructural complexity.

3.2.2 *Good technological capabilities:* The PI and some of the RAs had personal preferences for high performance software, especially to support text processing, statistical analysis and database management. In addition the increasing size of data files lead some staff to desire larger hard disks. Some analyses may take an extraordinary amount of time to run on the available computers, or may be accomplished only with more sophisticated software. The staff's desires for advanced technological capabilities conflicted with their budgetary constraints.

3.2.3 *Ease of use and maintenance:* While this value may be consistent with lower support costs, it may conflict with technological evolution. For the DCRP it has implied standardization in a common "core" of software and hardware for use by all group members. This strategy limits the flexibility of individuals to customize their computing arrangements to personal tastes. At the most basic level, "ease of use" may simply mean getting routine jobs done efficiently. But some research tasks — for example, preparing papers with data tables for final journal submission — may entail work that is far from "routine."

3.2.4 *Accommodating the geographical dispersion of team members between PPRO, ICS, and their home offices:* PPRO and ICS were located about 3 city blocks apart. The DCRP staff lived between 3 miles and 25 miles from the campus. This geographical dispersion set structural conditions on work life, and it has strongly influenced the DCRP's selection of computer tools, such as those which support text processing.

3.2.5 *Making the DCRP's computing arrangements congruent with those of the organizational units with which the team members most frequently interact -- PPRO and ICS:* Since PPRO and ICS relied on fundamentally different operating systems and applications software, compromises were not simple.

No one of these five criteria dominated in most of the key choices. But the DCRP found ways to resolve them into generally coherent computing strategies. The details of this strategy depended upon the mix of economic, technological, and human resources available to the research team and the researchers' tastes.

3.3 Computer Use and Support on the DCRP

Computerization projects are often dynamic. Their participants change equipment, the social organization of access to equipment, data, and computing skills over time. The term *developmental trajectory* denotes "the sequence of ... past social and technical configurations and the sequence of ... potential future configurations (Kling and Iacono, 1984:1219)." The hardware and software used by the DCRP's staff changed over time (Jewett and Kling, 1991). Moreover, the DCRP's staff participate in several academic units whose computing arrangements were also being changed along independent trajectories. Some key strategies of the DCRP's staff were negotiated in terms of computing developments *on the campus, in other academic units, and in PPRO* (Strauss, 1978).

In the 1970s the PI and his students computed on a DEC PDP10 mainframe system owned by the university's central academic computing facility. In the early 1980s, they shifted to a DEC PDP20 minicomputer owned by ICS. He purchased a PC with his own funds in 1983, but he and his students continued to use ICS' DEC PDP20 minicomputer for a substantial fraction of their work and communication via electronic mail.

ICS was moving along a developmental trajectory from reliance on central computers towards a collection of distributed computers running Unix and linked by ethernet. During the early 1980s, the department acquired DEC Vaxes and other Vax-clones which used Berkeley Unix. As part of a negotiation to sell a DEC PDP20 and to help the PI move from it an overloaded DEC VAX/750 minicomputer which ran Berkeley Unix, ICS donated funds for him to purchase his own microcomputers. The PI selected six IBM PC-XT clones with 20MB hard disks, several Epson dot-matrix printers, and key software. Apple Macs did not yet have a large suite of varied application software, and the PC's enabled the PI to build on his existing expertise. In 1987, the PI purchased a -286 PC clone and a color monitor with his own funds when he found that relational databases ran slowly on the XT's.

Starting in 1988, the ICS department began making -286 clones with EGA monitors which were part of an equipment gift from a computer vendor sporadically available to the PI. The staff found their speed more satisfying, even for routine tasks like word processing. Because computer science faculty preferred engineering workstations or Apple Mac-IIs, the PI was able to obtain 8

PC/AT clones with little contentiousness in 1989. He gradually replaced individual computers —- sometimes one at a time and sometimes in groups. All of the DCRP's machines are PC-AT compatibles (one with a "-386" processor). In 1989, the project modified some of the "-286" machines with additional disk drives that were "cannibalized" from the XT clones. In 1990, they again increased the disk storage capacity of several computers to 70MB by swapping for other PCs owned by ICS. At the same time they upgraded the video displays of a few machines to the "VGA" standard.

3.3.1 Document writing, formatting, and production

The DCRP uses software for writing, data analysis, and record keeping. We could illustrate the way that managing infrastructure influenced the choice of any of the DCRP's database or text processing software. We have chosen to focus on text processing since the issues should be more accessible to readers. Moreover, many professionals (except perhaps for lawyers, school teachers, and senior managers) and clerks in the US use computers to write, so the tasks and issues are now commonplace.

A central focus of the DCRP's computer use is preparing documents such as letters, memos, field notes, scholarly articles, and questionnaires. Between 1985 and 1990, the DCRP made substantial changes in the capabilities for producing documents. Between 1985 and 1988, the project staff drafted and edited documents on PCs and printed most of their drafts on dot matrix printers owned by the project. The PI formatted final drafts for many of the DCRP's publications on a laser printer owned by ICS. After 1988, the DCRP staff found ways to make laser printing more accessible for a wider variety of memos and article drafts.

During the 1980s laser printed papers became increasingly common in the DCRP's research world. Laser printed papers became a signifier of high quality professional work and of research quality. The project staff had three major strategies for producing laser quality documents, such as questionnaires for surveys and papers for professional distribution. One strategy was to subcontract the final stages of editing and printing to PPRO's office staff. They gave their text files (on floppy disk) formatted in the DCRP's word processing formats (e.g., PC-Write until mid-1988, and then WordPerfect 5.x). PPRO staff translated these files into their word processors format (Microsoft Word 3.x) and then reformatted them. PPRO administrators continually encouraged DCRP to use their staff for office services.

But project staff found continual annoying problems. Since DCRP and PPRO's text-processing group had incompatible word processor formats, the project staff faced costly and tedious reformatting when they wanted to "take a document back" after passing it to them or they were faced with having the PPRO staff recharge the project staff for all subsequent revisions. Further, the PPRO staff were reluctant to purchase a word processing conversion package that would translate files between PC-Write and MS-Word or between WordPerfect 5.x and MS Word[3]. This was a particular problem with documents which were revised in several discrete rounds of review, such as journal articles.

In addition, the DCRP staff sometimes found turnaround time to be excessive. The DCRP staff varied in their text processing skills. But professional office staff were required to learn MS-Word on their own, using Microsoft's cumbersome manuals. The best of the office staff produced high quality work, but they were reluctant to tightly supervise the work of junior workers who often had lower work standards. For example, errors often crept into revisions of data tables, requiring additional auditing by the DCRP staff. The PPRO office staff would cheerfully correct their errors, but recharged projects for their additional time. But the most important problem, in their view, was the high overall cost of PPRO's production services. The DCRP's interest in cost containment led the PI and RAs to seek alternative ways to laser print complex documents, such as journal articles and questionnaires.

A second strategy was to print on an Imagen laser printer in the ICS Department. This solution involved a different set of dilemmas. The laser printers were driven by a complex, command-driven mainframe-based formatter (Tex or troff). Computer scientists often prefer formatters like these because they allow fine control over document layouts, support the printing of complex mathematical equations, and are customizable. But they also require significant time to learn and a taste for programming; documents formatted with them can be easily garbled with a misplaced character. The PI developed good skills in using Tex, even though Tex's use was doubly cumbersome[4]. Once a document was modified to include Tex's special commands and delimiters, these would appear mixed into the document if it was printed with the DCRP's PC-based text processors[5]. The DCRP's senior research associate at the time was reluctant to spend time mastering Tex. Because she wrote at her home, about 25 miles from UCI, and she had no consultant readily available to help with Tex's idiosyncrasies. Worse, she had no way to preview the effects of the formatting commands that were embedded in a document. In occasionally using Tex, she made many trips back and forth between a printer at ICS (to pick up output) and the PPRO office or her home about 40 minutes away from campus to fix problems in the output.

The DCRP's third strategy before 1988 was to use a PC-based print enhancement program, Lettrix, to refine the visual quality of the dot-matrix printouts. It worked with the DCRP's primary text editor (PC-Write 2.x) and could be used at any of the DCRP's locations. However, it was painfully slow for long documents, since it could take well over three minutes to print each page (e.g., two hours to print a manuscript as long as this chapter). Moreover, since PC-Write occasionally misformatted footnotes when it was used with Lettrix, the PI and key RA's spent some grueling hours reformatting and reprinting long papers close to publication deadlines. Lettrix helped the staff use PC-Write to produce more professional looking documents on a dot matrix printer, but it was far from "what you get is what you expect."

The capabilities of PC-based word processors have continually improved, and by 1987 some of the technically most capable word processors began to include better capabilities for handling footnotes than the word processor (PC-Write 2.7) used by the DCRP. A key advantage of PC-Write was that it was very fast in response time on any of the DCRP's 8088-based PC's, including two with floppy disk drives. In 1987, the PI replaced these floppy drives with 20MB hard disks and made a wide variety of somewhat slower but more powerful word processors more attractive. The PI directed a team of students in exploring the possibility of shifting to one of two major word processors: Microsoft Word 3.x and WordPerfect 4.x. The DCRP staff did not see adequate advantages to either one. In 1988, WordPerfect 5.0 was enhanced with new features which allowed formatted documents to be previewed on screen at any of the DCRP's locations, and to generate Postscript files which could drive ICS' laser printers without resorting to Tex. While PPRO's office staff continued to use Microsoft Word, the advantages of previewing and simplified laser printing at ICS were major incentives for changing. The PI also found the newest version of Microsoft Word to be clumsier for some key operations, even though it was a powerful and popular package[6].

The combination of the DCRP's new WordPerfect 5.0 word processor (with preview capability) and new Postscript compatibility for ICS' Imagen laser printer enabled the project staff to significantly reduce their lengthy trips to ICS to preview laser printed documents. But the DCRP staff could still not eliminate all trips to ICS. They experimented with using an Apple laser printer used by PPRO's office staff which accepted Postscript files. PPRO wanted the project staff to give them files to print and to recharge both labor time and equipment use. The DCRP staff wanted to use it as a walk-up self-service with charges for equipment use, akin to the photocopier. Moreover, the DCRP staff wanted to be able to use their own word processor on one of PPRO's PCs to make quick changes to documents if there were unexpected formatting problems. However, PPRO's office staff had their postscript-compatible laser printer located inside their carrels and attached to PCs and Macs which they used for ongoing work. The PPRO staff was moving to Macs, and shifting the printer back and forth between PCs and Macs required rebooting some systems. Consequently, they found the DCRP's attempts at self-service to be intrusive, and the DCRP staff were barred from continuing.

In the summer of 1989, the PI negotiated with the ICS department for two Hewlett Packard laser printers (in exchange for a complex administrative assignment). During the first year with the laser printers, some staff became seduced by strategies to improve their usefulness. When DCRP staff used ICS' laser printer with Postscript output, the PI became used to the flexibility of arbitrary font sizes (e.g., 11.2 point). This font sizing flexibility was particularly helpful in formatting documents to be appear in as legible a font as possible while fitting within some designated page limit, such as 1, 10 or 25 pages.

Some undergraduate students and one of the graduate research associates spent substantial time working to create a larger selection of font sizes than are provided by cartridges through the use of soft fonts. Some of the staff spend time "tweaking" the output based on new capabilities: would the headings look better in bold-face or italics? ... or maybe large print or a different type face? ... can the DCRP staff meet a page count with 12-point or 11-point type, or should it be 11.5 point? ... maybe a little more or less spacing between the lines? ... how about adding a border around the tables? ... *ad infinitum*. It's fun, seductive — and often took several hours more than planned.

Increasing computer capabilities gave the project staff a great deal of control over the final appearance of documents, and enabled the project staff to meet page limits and format specifications for conference and journal submissions quickly with minimal marginal expenditures.

3.3.2 Staff Training

All of the DCRP staff arrive with some computing skills. Many of the graduate students and undergraduates own PCs. When they join the project, they usually do not know the particular suite of application software, and usually they aren't sufficiently sophisticated to easily manage the complex array of project files. For example, the DCRP maintains dozens of different kinds of manuscripts, memos, databases, and charts in a myriad of topical subdirectories. Most undergraduate students use subdirectories very crudely since they usually manage a few papers and write a few programs. Consequently we often have to teach them more sophisticated ways to manage and catalog complex sets of files, as well as the use of software to help manage a complex disk organization.

The DCRP staff teach research substance and process in the course of the work. But the DCRP staff also teach a significant body of computer skills to new members. The undergraduates -- upper-division computer science majors -- often require extensive training in the specific software. They also require training in practical procedures to effectively manage software and data in a group setting -- such as carefully labeling diskettes and informing others about ways that they reconfigure PCs.

The project ability to develop infrastructure is limited by its size, budget, and the many competing time demands. The DCRP staff design jobs around the available staff, rather than trying to force-fit individuals into duties for which they are ill-equipped or in which they have little interest. The result is a pooling of expertise which emphasizes complementary skills and cooperative effort; the DCRP staff help each other. The DCRP staff can usually find a group member who has encountered a problem before. It may take a phone call or an email message, or it may have to wait until the weekly meeting, but there is usually an answer available.

The DCRP staff also maintain an extensive library of software documentation: vendors' manuals, third-party tutorials on the major programs, and about 120 pages in six tutorial manuals in which the DCRP staff have characterized their key software and procedures. The simple availability of these references is not sufficient: staff members must read them and have the skills to understand them. Project members vary in their diligence in reading the documents. Refining these customized manuals involves a tradeoff: staff time to write and update them versus learning time saved for new group members (or learning time available to read them).

The DCRP staff try to adopt software for which there are good books sold by trade publishers. This makes it easy for undergraduate student assistants to buy instruction manuals and to learn the software with less administrative complexity. In particular, there are numerous books for the DCRP's primary word processor and relational data base.

Finally, the DCRP staff use informal one-on-one training, conducted by either the PI or one of the senior RAs. At best, individualized tutorials allow the project staff to tailor training to the specific needs of each individual. At worst, the time the DCRP staff invest in new undergraduates might not pay off, if their class schedules and outside work commitments limit the time they can devote to this research project.

In order to contain costs, the DCRP staff try to maintain their own hardware and to perform minor upgrades, when it is possible. Some changes may require internal modifications to the computer (e.g., adding boards or wiring, resetting switches), and not all users will be willing or equipped to undertake these tasks themselves. The DCRP staff usually evaluate the potential gains of an upgrade against the direct expenses and the likely staff time required to fiddle with equipment. But even seemingly straightforward modifications can involve unexpected work.

Learning and competence building are continuous processes for all the DCRP staff for several different reasons. It takes most staff about 3 months to develop adequate competence in the word processor, database, analytical packages and utilities which they use for their primary tasks. The undergraduate students often explore new software, and some of their experiments lead to adoptions by the DCRP. The DCRP usually upgrades key software to be able to exploit new features, and the upgrades involve some new learning. Periodically, the staff finds system anomalies, and debugging them often leads to some new minor skills. Finally, staff who are more experienced in a particular package are often advising less experienced staff, and both learn in these interactions.

3.3.3 Computer system administration

Maintaining compatible configurations on all of the eight DCRP PCs is not a trivial task. The DCRP staff rely on an ICS undergraduate "system administrator" to provide support for the computing milieu in exchange for special study course credit. Designing this job is a delicate balance so that it has legitimate academic learning value to the students and also helps leverage PI and RA time. This student normally devotes about ten hours per week to these activities, and works under the supervision of a graduate RA from ICS or the PI. Most system administrators have worked for the project staff for two quarters, since it takes them much of one quarter to effectively learn the computing environment. Occasionally the DCRP staff have two students working together. Other project staff provide varying kinds of computer support (e.g., training) depending on their interests, time, and skills. The PI channels key support activities, such as exploring new software, installing upgraded software, performing routine maintenance and repairs to the unpaid undergraduate assistants. It is doubtful that the strategy of computerization examined here would have been effective without the availability of skilled undergraduates.

Support is the hidden soft underbelly of computerization. The DCRP staff have estimated the overall time required for computer support and maintenance within the DCRP to be in the range of 700-850 hours per year. This estimate assumes one undergraduate per quarter (350-400 hours/year), 5-7 hours per week of RA time (250-350 hours/year), and 2 hour per week of PI time (100 hours/year) -- and these figures seem very conservative. The distribution of hours may vary depending on the available staff. A skilled undergraduate with a high level of personal initiative will reduce the time that graduate RAs spend on computing support tasks. In turn, a graduate RA with well-developed computer skills and interests can perform many support and supervisory tasks that would otherwise fall to the PI. The total time which all staff members devote to the project is approximately 5000 hours/year. Therefore, computing support consumes at least 15% of the DCRP's time budget -- a significant commitment.

4.0 SITUATIONS, CONTEXTS and PROCESSES of COMPUTERIZATION

Before embarking on a substantive analysis of computing developments, we need to lay our conceptual groundwork. A major difference between discrete-entity and web models hinges on the concept of "context." An operational definition of *context* can help us. We have used the lay definition of "context" as "something larger" that gives meaning to the topic in focus. This is a rather vague conception for empirical analysis. Our operational definition will hinge on another common and vague term - "situation," which we will also sharpen. We will also explain how our analysis rests on two different perspectives on social action in organizational life: organizational process (Cyert and March, 1963) and negotiated order (Strauss, 1978).

4.1 Situations of Computing Development and Use

Our primary unit of analysis is the *situation* in which individuals or larger collectivities carry out their going concerns while they engage with computer-based technologies. Any account of computing development or use implicitly defines situations by identifying participants, equipment, location, and time scales. Not all situational definitions are equally useful for understanding the actual consequences of computerization. Situations should be defined by set of participants, equipment, spatial and temporal relations, and social processes which shape participants actions.

Situations can vary considerably in:

1. the *number of participants*;
2. the *set of artifacts* involved;
3. the *spatial scale and arrangements* of activity;
4. the *time periods* of key social activities; and
5. the *primary social processes* that shape critical behavior.

Particular situations may be located along the first two dimensions based on the number of participants or artifacts within them. Situations may be located along the spatial dimension by the amount of space their equipment occupies and by the amount of space their participants take up when engaging with computing and related activities. Situations may be located on the time scale based on the periods of time over which key events or social relations developed. At the smaller end of these dimensions is the fleeting encounter between a person and a discrete computing resource - e.g., an airline passenger and a video screen listing flight departure times in an urban airport. Somewhat larger on these dimensions is the use of an automated class assignment and scheduling system to match students, classes, and times in a university. Even larger still is the monthly preparation, dispersal, and reception of 40 million monthly individual payments by the Social Security Administration.

Situations are somewhat open-ended, and these structural dimensions and their characteristics are simply useful constructs, rather than eternal verities. It is not sufficient simply to identify a set of participants. Critical social relationships between them must also be identified. At the very least, the bases and kinds of cooperation or conflict between various groups is often critical. Computer systems will usually "work" more smoothly if all parties cooperate in sharing relevant data, cooperate in providing infrastructural support, etc. When a behavior setting that surrounds computer use is populated by groups who are in conflict, computing resources easily become bound up in the larger conflicts.

The kinds of social relations that bind participants together are particularly critical: their division of labor, relative status and power, etc. Processual elements of a situation will also be important: the beliefs of participants, their resources, common practices and procedures, and the rates of change of key social relationships. We will discuss both social relations and key social processes in a following section.

4.2 Context of Computing Development and Use

Discrete-entity and web analysts differ considerably in drawing boundaries which define situations for explaining the payoffs, problems, and dilemmas of computing. This section examines ways of drawing social boundaries for web analyses and contrasts them with the strategies of discrete-entity analysts.

Discrete-entity analyses use formal criteria for bounding the development and use of computer-based systems. These formal criteria -- such as official jobs and roles typically lead to compactly defined situations. For example, a dominant image in the computing literature emphasizes one person employing a particular piece of equipment as a tool in some well defined task. Some examples are the programmer working at a terminal developing a new application, the manager using a routine computer-based report to make a specific decision, the bank customer extracting funds from an automated teller machine are common examples, and the student using a special instructional computer package. Description of situations that are larger than one person and one machine are also common:

a. Larger population scale (e.g., The students and faculty at XYZ University (XYZ-U) have access to a network of 50 powerful workstations and 20 minicomputers for text processing and scientific computations to support scholarship and instruction.
b. Larger equipment scale (e.g., the computing equipment at XYZ-U supports several text editors (EMACS, vi), formatters (Nroff, Troff, LaTex), programming languages (C, Prolog, Fortran-77, and Ada), a database (Ingress), a statistical package (SPSS-X) and all the machines run Berkeley Unix 4.3.
c. Larger time scale (e.g., XYZ-U has owned this equipment for two years).

In these situational definitions, groups are usually identified by their formal relationships and behavior is often defined by official roles. In contrast, web analysts use fundamentally social criteria for defining situations. They ask: how do participants conceptualize their actions, what resources they have available, what are their common practices and procedures, what coalitions they participate in, what options they have, what constraints they face etc.? There is considerable evidence that social elements such as these profoundly influence the kind and quality of computer-based products that organizations produce, adopt, and use (Kling, 1980; Kling, 1987; Kraemer, Dickhoven, Fallows-Tierney, and King, 1985). Resources, relationships between interest groups, social meanings, procedural constraints, are often defined in another situation that is yet "larger" on some of the four dimensions of population size, equipment, space, and time. Any large organization that employs computer-based technologies is rich in examples of computing developments that are shaped by larger situational definitions that include the purposes, common practices, resources, procedural constraints, options, etc. that shape smaller scale developments. But one cannot easily and accurately infer these inherited meanings, social relationships, social structures and resources from the smaller definitions.

We use the term "context" to refer to these social relations and structures in a larger scale situation[7]. We often say that one situation (Sc) provides a context for understanding a second situation (Sf). The term *context* indicates a relationship between the two situations, Sc and Sf: Situation Sc is the level of analysis at which intentions which are observed in Sf are enacted and constraints made that influence other behavior observed in Sf. For example, the computing milieu described above at XYZ University would typically be more attractive to computer scientists than to social or physical scientists. Some faculty, such as economists, sociologists, physicists, etc. are likely to want software developed at other universities and which runs on IBM mainframes or on Vaxes under VMS, but not on the minicomputers at XYZ University under Berkeley Unix. The elements (a,b,c 2 paragraphs above) defines a situation Sf.

The relationships between economists at XYZ-U and their colleagues at other universities who expect certain kinds of time series analyses which are not supported by the software described in (b) defines part of a larger situation Sc. Since academic scholarship is evaluated in a national community, often funded by Federal agencies, and rewarded by promotions committees on particular campuses, these groups are likely to participate in the situation (Sc) which a scholar, like our economist, uses in defining the appropriateness of the computing equipment at XYZ-U. Sc is the context which makes the attractiveness of the equipment described in (a,b) above to an economist who is a participant in Sf.

It is always possible to enlarge the definition of a situation to larger and larger populations, sets of equipment, space, and time. One could examine computerization at XYZ University in the context of industrial development in late twentieth century United States. We need operational criteria which lead to useful and manageable boundaries. Below we propose three criteria which restrain any easy expansionist sentiment that aim for the most holistic defining situations; they require the analyst to make explicit causal ties between key elements of the defining situation and variations in the development of use of the focal computing resource.

Where and how should one draw useful boundaries? It is possible to set up boundaries a-priori. Turner (1981), for example, suggests that the whole organization that develops or uses computing provides useful situational boundaries. This criterion is a useful heuristic and expands the range of participants, equipment, space, resources, and procedures to a larger set than are usually found in most discrete-entity analyses of computing developments. Unfortunately, it does not help explain patterns of computer use that are influenced by social or economic patterns in still larger situations like Sc above (Kling and Gerson, 1977; Kling, 1978a-c; Goodman, 1979; Dutton, 1981; Danziger, Dutton, Kling, and Kraemer, 1982; Dutton and Kraemer, 1984; Kraemer, Dickhoven, Fallows-Tierney, and King, 1985;McHenry & Goodman, 1986; Jewett and Kling, 1991). In these studies, organizational participants select computing arrangements to leverage their negotiations in a larger social order. They also find that participants are constrained by resources and organizational routines that are defined outside the formal boundaries of their organizations.

Organizations that have critical on-going negotiations with outsiders -- clients, auditors, regulators, vendors, competitors -- will sometimes develop computing arrangements to enhance their bargaining positions. To understand these choices, a larger situational boundary that includes this expanded organizational set must be drawn. These boundaries cannot be completely defined a-priori. Nor can they be defined before the participants of these larger negotiating contexts are identified. Also, the choices that organizations can make relative to computing are limited by actions taken by agents outside their boundaries (e.g., clients, equipment vendors, auditors, competitors, labor unions, regulators) or by the ecologies from which they draw resources (e.g., local labor markets, vendor supply practices). These critical exogenous elements will vary from situation to situation, and fixed boundary criteria will be far too rigid for explanatory purposes. Useful boundaries depend upon the behavior being explained and the lines of explanation adopted.

We propose three criteria for drawing the boundaries of larger scale situations Sc which serve as useful contexts for a given focal situation Sf (Kling, 1987). The *defining situation* Sc is bounded by the populations, equipment, spatial and temporal (PEST) elements that meet either of these criteria:

C1. the actors in Sf *depend upon* the PEST elements of Sc when acting in the focal situation Sf; *Or*

C2. the actors in Sf *take account of* the PEST elements or time of Sc when constructing their actions in the focal situation Sf; *Or*

C3. the PESTs of Sc *constrains the action* of the actors in the focal situation, Sf. (The focal actors need not be aware of their social relationships with the PEST of Sc and their influence on their behaviors.)

The defining situation Sc also includes those social processes that meet criteria C1, C2, and C3. These three criteria are essentially social. In contrast, discrete entity analysts usually draw formal boundaries and stay within one level of analysis. Web analysts almost by definition work with multiple units of analysis, both Sf and Sc at minimum. Web analysts choose boundaries for the defining situation Sc depending upon the questions they ask. It takes care to draw boundaries for a defining situation Sc that are neither too limited nor so encompassing as to explain nothing.

Suppose one wishes to answer the question: how useful will a particular text processing be in helping the Desktop Computing Research Project (DCRP) Staff? One way of answering the question is to examine the mix of letters, memos, and papers produced by the DCRP staff and examine the ease of producing them given the features of a given word processor. A web analysis goes beyond this kind of task assessment by examining the kinds of resources available to participants. In the DCRP case, criterion C2 asks, who do the project staff take account of when they are writing and disseminating documents (e.g., recipients of DCRP correspondence, respondents to questionnaires, readers of technical reports)?

Laser printers were a key resource which the DCRP staff did not own before mid-1989. But they could use laser printers located at ICS or at PPRO. (Each printer was available on a different schedule and under different conditions). Criterion C1 identifies the printing resources and people who control access to them in both ICS and PPRO as part of the situation of text processing use on the DCRP. Criteria C1, C2, and C3 should be applied to develop a suitably "large" defining situation, while avoiding unworkably "vast" situations. There would be no explanatory gain in enlarging the defining situation to include "advanced industrial societies since 1900." Fortunately, criteria C1, C2, and C3 will lead to a focus upon the kinds of participants and social relationships examined in the last paragraph. C1, C2, and C3 will exclude improbably wide situational boundaries like "advanced industrial societies since 1900" since people who are long dead are very unlikely to have much direct influence upon an RA in the DCRP (C2) or be taken into account by her (C1). Such a situational definition helps focus on the information processing leverage provided by the text processor. The temporal scale should be closer to the life-cycle for the software rather than the time frame for the RA to write on memo.

4.3 The Political and Interpretive Dimensions of Organizational Action

So far, we have emphasized relatively static dimensions of situations: populations, equipment, space, and time. But to breathe real life into situations that develop in and around computing, we need a model of the social terrain on which people act (with or without computers). Modern complex organizations have a special kind of terrain: they produce most of their goods or services for some external clientele, and organize their work through a variety of specialized groups and explicit divisions of labor.

We will treat organizations as naturalistic open systems (Scott, 1987:20-24,100-101), and will explain these terms below. There are numerous sociological models of the ways that organizations behave. But the information systems literature is dominated by "rational systems" models. In the extreme rational models, writers characterize organizations as unified task systems which their uppermost managers guide subordinates towards clear goals through explicit strategies. These models do not suggest that there is much ambiguity in the kinds of problems facing organizational participants, their goals, lines of action, available means, etc. Charles Perrow has written eloquently about the way in which these accounts reflect a managerial ideology which attributes significant insight, foresight and power to managers (Perrow, 1986).

But strong assumptions of rationality are also contained in other, more behaviorally descriptive models of organizational behavior. One of the more influential of such models is the *organizational process* model (Cyert and March, 1963; Allison, 1971; Steinbruner, 1974). In this model, organizations are noisy production systems composed of a network of work groups which exchange and produce goods and services according to a host of relatively standardized procedures (SOPs), formal and informal. Information, actions, and material flow imperfectly from node to node along standard channels. The characteristics of channels and nodes are also subject to routinized temporal patterns: staff can change at specific nodes because of shift work each day, regular vacations, or rotating assignments. Communication and transportation through channels is also subject to temporal regulation -- material may be transported on fixed schedules, information passed at regular intervals, etc. Certain key events and resources may occur on regular schedules: systems are often audited, data archived, and budgets renewed on regular cycles. Channels and nodes are imperfect: delays and losses commonly occur.

In this model, organizations and their subunits are characterized by their routine outputs and their average behavior in providing these goods or services. This model emphasizes the production of typical outputs and the character of typical inputs. Clients make typical demands and organizational participants develop routines (formal or informal) to get on with their work. Resources come from relatively standardized sources (e.g., standard vendors, standard labor pools, etc.). Outputs go to standard markets, etc.

The organizational process model emphasizes the purposive side of organizations, since it focusses on tasks and goals, rather than the construction of meaning and symbolic action. It departs from the most formal rational approaches in many critical matters: "organizational goals" are redefined by subunits that enact them, organizational participants have limited capacities to process information; participants' pursuit of their interests is tempered by their propensity to avoid uncertainty; participants have limited capacities to process information; behavior is characterized by many common routines and procedures, even if these don't match formal or public rules and practices; organizations adapt slowly as a byproduct of solving particular problems (Cyert and March, 1963). Groups neither share common values nor do they agree on common goals, but jointly cooperative outcomes are common.

In the organizational process model, larger organizational arrangements, the values of participants, the distribution of rewards and constraints, and the set of procedures which describe ongoing behavior are largely taken for granted. We require a different model to ask how these elements of organizational life develop.

In the rational models of organizations as collectivities devoted to specific purposes, organizations should gracefully dissolve when they meet their goals. There is an interesting body of sociological research which shows that organizational members often develop new purposes to keep an organization going, even when original goals were met or at least resolved. Rather, many key members may have a key stake in keeping the organization stable or growing. The label "naturalistic" refers to the practices of organizational members in keeping their organization "alive" -- like an organism in nature.

An interesting example, today, is the U.S. military, in the wake of the dissolution of the Soviet bloc. Since the late 1940s, the US military justified its huge budgets in terms of fighting communism, or being able to fight a war with the Soviet Union or China, if necessary. With these military threats significantly reduced, one might expect the US military to shrink substantially in size and cost. However, if the US military acts as a "natural system," its officers will try to maintain its size and influence as best they can.

Another kind of example are the central information systems departments of large organizations. Through the 1970s, centralized data processing departments often maintained a relative monopoly over the development and operation of organizational information systems. In the 1970s, the lower cost of minicomputers began to make departmental centers financially feasible. In the 1980s, the rise of desktop computing has moved significant computing into workgroups, and further eroded the demand for centralized data processing departments. The natural systems models view the members of organizations at all levels as acting on behalf of their self-interest, not just simply trying to generously solve organizational problems. This does not mean that the staff of centralized data processing will embrace all possible sources of future services. Rather, it means that they will try to assess possibilities for service in terms of what they think that it means for the growth of their department, as well as what it means for other departments in the organization. For example, a centralized IS department may try to insist that they will build a significant new information system rather than recommending that a department be able to promptly purchase a better one from an outside vendor (see, for example, Jewett and Kling, 1990).

We also view organizations as open systems. In closed system models, organizations are not much influenced by their environment. They are a coherent social unit oriented toward the pursuit of specific goals. In open system models, organizations are "coalitions of shifting interest groups that develop goals by negotiation; the structure of the coalition, its activities and its outcomes are strongly influenced by environmental factors (Scott, 1987:23)." In the DCRP case, the research strategies and preferences for laser printing were influenced by practices in the teams' research community *outside* their own university. Similarly, the computer scientists preference for Unix and related software was based on emerging standards in academic computer science, rather than only on a selection of the most efficient equipment to carry out their work.

We view organizations as work systems in which participants make decisions about their work rather than as decision systems in which work is incidentally done. For example, writing papers entails physical work, and is not just a physical byproduct of a stream of choices. This distinction is of far reaching importance. While many people make decisions in the course of their work, most jobs result in a set of actions based on whatever decisions are made by the actor or others. In organizational life, the physical work with and around computers influences what is done and what is used. We take seriously the lines of action in which people come to engage with computing, the going concerns of the organizations they inhabit, and the patterns of incentives and constraints which influence them (Kling and Scacchi, 1982). In this chapter I will focus on two naturalistic, open system models of organizational behavior, *negotiated order* models (Strauss, 1978) and *institutional* models.

4.3.1 Negotiated Order Models of Organizational Action

Negotiated order analyses focus on the ways that organizational practices, such as systems standards, are worked out between groups with differing interests and orientations. Institutional analyses focus on the ways that certain practices, such as commitments to specific operating systems or programming languages, are hard to change, even when they are not very effective. These two naturalistic open systems models complement each other.

The negotiated order model views the actions of organizations as the byproducts of *ongoing* negotiations. Analysts who use the negotiated order model have often taken the coordination of complex organizations as a central issue, and one which is problematic for participants. Analysts who adopt an organizational process analysis usually assume work is factored into tasks for specialized subunits, and that dominant coalitions play a strong role in assuring that organizational outcomes do not undermine their interests. Negotiated order analysts do not take the influence of a dominant coalition for granted. They pay special attention to who is bargaining over what, and with what options in different negotiating contexts. Both models presume that participants' definitions of situations are influenced by their socialization, training, group allegiances, and the social worlds in which they participate.

The organizational process model identifies the key concept of standardized operating procedures (SOPs). But it takes these routines for granted and simply assumes that they were the outcome of some bargaining between coalitions; but the bargaining takes place offstage. The negotiated order model focuses on the creation of procedures and how variations are negotiated. While both models accept conflicts as an ordinary element of the ongoing interplay of groups and organizations, the details and tapestry of conflict are more center stage in the negotiated order model. Participants' strategies, the stakes they contend for, their options and relative power, and the bargains they strike in a negotiating context are all central. Moreover, participants in some complex work organizations engage in multiple and overlapping negotiating contexts in which their stakes and option provide cross subsidies and constraints (Becker, 1960). (For example, in the case of the DCRP, the faculty PI and the RAs participated both in the research institute, PPRO, and in academic departments.)

4.3.2 Institutional Models of Organizational Action

Institutional analyses hold that organizations may not change, even when they would improve their resources, power, etc. in response to shifting dependencies with other participants or their "environment." Interdependent computerized systems have important "institutional" dimensions. "Institutions" are organized social arrangements which persist and are taken-for-granted by participants. The organized technical and social elements of a computerized systems have important institutional dimensions when they are not all fluid and easily subject to change by specific coalitions or interest groups. The inflexibility of some organized computing arrangements may impede those who desire large-scale changes. When routines become outmoded and no longer useful to the organization, many managers and users may push for large-scale changes in their computing arrangements.

The organizational routines which facilitated efficient activities and stable environments in the past may prevent the changes which key participants may believe are critical to the organization's continuing survival. We call this kind of organized rigidity, "institutional inflexibility." Rather than focusing blame for failed implementations on resistant end-users or on "insensitive" system designers, the focus is on the social arrangements and practices which support the development and operation of the computerized systems.

The payments system of the United States federal Social Security Administration (SSA) illustrates institutional inflexibility. It produces over 40 million checks per month and was developed in the 1950s in a crude programming language for specific computers (Autocoder). And many of the old programs remained relatively intact through the early 1980s. The SSA has tried to overhaul the payments system at least three times in the last 15 years without success, although a new Systems Modernization Plan that may be completed in the 1990s is in progress (Laudon and Laudon, 1991). Even though some new laws have been passed during the late 1970s (e.g, eligibility requirements) which change the way in which payments are legally distributed, none of these laws had actually been implemented in software for at least five years. We do not have adequately detailed accounts of the earlier efforts to change the SSA's payments systems to understand where the difficulties lay. But the persistence of the SSA's outdated payments system architecture from the 1950s indicates that large scale computerized systems can be exceptionally difficult to replace.

Examining the institutional dimensions of computing infrastructure is important because it helps an analysts identify which technologies and practices are most difficulty to change. **The actual usability of computerized systems in specific social settings is the critical factor in assessing social benefits**, not the potential capabilities of the technology as it may be used by special individuals or groups. Ironically, computerized systems that are well-used and have stable infrastructures for supporting and using them will be usually much more difficult to change or replace than those with less associated social structure and fewer participants.

When analysts emphasize the social and political choices that organizational actors have made over time, they are placing its institutional character in the foreground. Many of these institutional dimensions are intangible and taken-for-granted. Long after some interest groups have lost power or influence in the organization, their interests and visions may still be embedded commitments to equipment, SOPs and organizational arrangements.

"Institutionalized behaviors" can vary between departments in the same organization. For example, accounting departments are much more likely to value accurate data, while production departments in manufacturing are more likely to value meeting quotas by regular shipping dates. Academic computer scientists are more likely to prefer programmable, powerful, and complex text formatters while people who do not have technologically sophisticated backgrounds may prefer simpler menu driven systems. These groups participate in social worlds which have different systems of meaning through which they value different kinds of technologies.

From an institutional perspective the ways in which work is organized across work groups in an organization influences the SOPs of participant groups. Different user groups often share some interdependent work schedules and routines. In manufacturing work environments, products are shipped as the result of the collective activities of many diverse departments and divisions. The interdependence of work routines implies that the contribution of some component, policy or practice hinges on its dependency upon other components, policies and work-group practices. For example, when a fast machine replaces a much slower machine, one expects speedier production. However, increased demand on the new machine may result in some groups of users waiting in line for access. Their lost waiting time may exceed the gains of speedier technology. Further, when the speedier equipment does expand productivity in one work group, it can lead to log-jams elsewhere. Thus, improvements in some components of an ensemble of computing technologies do not necessarily improve overall performance.

When computerized systems are characterized as institutions, organizational politics play an important role. Powerful groups can attempt to influence the developments of a computerized systems even when they can not control the outcomes they desire (Kling and Iacono 1984b). Data Processing (DP) departments specialize their work over time as they routinely fulfill the computing preferences of particular groups over the preferences of other groups. They are likely to hire new staff who primarily share their world views -- in preferring particular application domains (e.g., finance), languages (e.g., COBOL), equipment vendors (e.g, IBM), etc. However, specialization and routinization reduce the ability of DP departments to engage in work activities which depart from their skills and SOPs at a later time. Consequently, an engineering department which seeks computing support for computational programs written in Pascal or C which run under Unix on a DEC-Vax will usually find systemic difficulties in obtaining meaningful service from a DP shop which develops financial applications on large IBM 308x mainframes under VM in COBOL. (The reverse is also likely -- that finance staff would have difficulty in obtaining high quality data processing service from an engineering oriented computing staff who prefer different kinds of applications, programming languages, and machines.)

Institutions develop a character based on the interests they have served in the past, and the world-views which bind their participants together. Incremental changes in computing arrangements usually improve the fit between a computerized systems and its organization. As a computerized systems fits an organization better, it is much harder for powerful actors to easily replace it. Participants organize their work lives around the belief that activities which have become routinized will persist. They come to depend on existing social and technical arrangements for working and for achieving personal goals. For these users, and for other important actors in the organization, the particular social organization of a computerized systems can become indispensable.

Staff stabilize and routinize their work environments and create highly organized computing milieus in the process. There is an important paradox: a useful and stable computerized systems can be problematic when key organizational actors try to change it in adapting to new conditions (Kling and Iacono, 1989). Even when computerized systems are fairly primitive and social relations are informal, elements of the computing environment can became highly organized and taken-for-granted. Critical social dimensions that can been taken into account by organizational actors include: the skills and work expectations of computing staff; the work prioritization schemes that had developed between computer staff and end-users; who controls new developments in the computing environment and the deployment of resources; organizational practices about training users and computer staff; the degree of specialization in terms of computing software and hardware; and past commitments to the specific work groups which effectively exclude other groups from decision-making.
Computing arrangements that are highly institutionalized will arise when there is shared control and competing interest groups vying for the resources of a computerized systems. An institutional analysis is critical for understanding how computerized systems are likely to be used in more complex work settings where constraints and limitations are part of the everyday work world. It is a paradox that the more stable and usable a current social organization of computing is, the more difficult it will be to make substantial changes. This does not mean that implementations of new computerized systems are impossible. Rather, the social organization of computing must be taken into account prior to the setting of realistic expectations of performance and schedules. In practice, we observe that new generations of computerized systems and other replacement computer technologies are often implemented and effectively used on a much slower time scale than many key participants expect.

Institutionalization has many ironies. Innovators like to see their innovations widely adopted for indefinite periods. Moreover, there are many economies of scale when innovations are institutionalized (Kraemer, Perry, King, and Dunkle, 1988). But the technologies and innovative practices of one era have often acted as a brake on subsequent innovation. For example, once an organization develops technical standards, they often remain in place for decades, despite the availability of technically better alternatives. COBOL is still a mainstay business programming language after 30 years, despite the wide availability of fourth generation database languages. MS-DOS and IBM-PC compatible micros remain the business standard, despite the availability of MacIntoshs with easier to learn interfaces and IBM's promotion of a more complex OS/2 operating system. In cases like these, the immense installed base of earlier technologies means that substantial conversions are also very costly, in changing equipment, skills, visions of "appropriate computerization," and organizational practices. We would be surprised to see COBOL and MS-DOS comparably entrenched in 30 years, since organizations do shift their technical commitments.

Similarly, the social organization of computing shifts over time. The widespread use of microcomputers has moved the locus of control of some systems from central information systems departments to end users. These shifts of control are coupled with shifting costs for computing and also changing professional ideologies about what computers are good for and how computerization projects should develop. But such changes are relatively slow in many organizations. Social theories of computerization have to accommodate social processes of change and of stasis[8].

5.0 COMPUTING INFRASTRUCTURE AS AN ORGANIZATIONAL PHENOMENON

In this section we examine the infrastructure of computing as a central feature of systems in organizations. Specifically, we will examine 1) how *the organizational relationships in which computing developments are embedded* influence their relative workability for participants; 2) the *co-requisites of computing developments*; and 3) the constraints placed upon developers and users from *historically accumulated commitments*.

When automated information systems are operated "routinely," the array of supporting resources strongly influences the quality of computer-based service which is actually available to users. The infrastructural resources for a given activity refers to those adjunct resources which help carry out the activity smoothly. Urban planners refer to such physical basics as roads, sewers, and utilities, which support social and business activities as the "urban infrastructure." Infrastructural resources for using reports from a computer-based information system include skilled staff, accurate and complete documents, sharp operations procedures, and enforceable equipment contracts. For software developers, infrastructure includes programming skills, information about systems, working hardware, etc. For computer hardware, critical infrastructural resources also include physical systems such as reliable "clean" electrical energy and low-noise communication lines[9].

5.1 Identifying Infrastructural Resources

Discrete-entity analysts can assume that infrastructural resources are available as needed, so they often overlook important resource gaps. Web analysts try to account for the actual bundle of supporting resources available to different participants, so they need a way to identify them. We identify key infrastructural resources by following the chain of resources, equipment, consumers and providers which support a given computing resource. This network is called the "production lattice" for the focal computing resource (Kling and Scacchi, 1982; Kling, 1987). Those resources that each consumer/provider depends upon constitutes an element in the infrastructure for the focal computing resource. We have found three ways to identify the production lattice for a particular computing resource:

PL1. Trace the chain of social and technical interactions from the focal computing resource outward to various suppliers of basic resources and skills which enable it. These basic inputs include data, supporting equipment, energy, communications, etc. They also include participants who provide expert assistance, ensure that basic inputs of adequate quality are provided by negotiating with suppliers, training resource producers, auditing the quality of inputs, etc.

PL2. Include those inputs or resources whose alterations yield (large) improvements in the quality of output provided by the focal computing resource.

PL3. Include those inputs or resources whose failures reduce the quality of the focal computing resource.

Criterion C2 (from Section 4.2) guides how far outward from the focal computing resource one should follow the chain of dependencies indicated by criterion PL1. Narrow boundaries drawn close to the computer room miss the dependencies of computerized information systems on other organizational resources such as inter-office mail and telephone lines. Narrow boundaries miss relations with equipment vendors or outside consultants which may also be critical. (See also, Kling, 1987)

==

Table #3 About Here

==

Table #3 lists some typical infrastructural resources for computerized text processing in the Desktop Computing Research Project. The infrastructural resources are those things which help various actors use the text processor and which support production lattices defined by criteria PL1, PL2, and PL3. The elements of Table #3 are primarily static, and we will examine three static characteristics first:

a. Infrastructural resources are layered. The focal computing resource for one participant (e.g., the operating system as seen by a systems programmer) is infrastructure for participants further up the (production) chain. (Infrastructure is **a relationship** between a focal resource and a supporting resource.) The computing infrastructure in organizations has often become more complex with the spread of multiple systems with different components in different organizational units and systems connected through communications networks. The communications wiring diagrams often just hint at the social complexity of the infrastructure for specific applications which draw data from one system, process it on another, and print it on third. When system components are owned by different departments and located in different physical locations, practical usage is sometimes complicated.

b. Infrastructural resources for computing may be coextensive with or part of other organizational procedures and resources. At various points, the mail room which disseminates reports, the systems which provide electrical power and air conditioning, procedures for hiring and firing staff, or for procuring equipment, can be drawn into the support of a focal computer system. In the case of the DCRP, the PI recruited some computer science undergraduates to learn about the management of a decentralized computing environment and to help administer the projects PCs in exchange for course credit. But these students' availability was influenced by demands from their other courses, including somewhat predictable examination schedules.

c. Some of the resources are relatively concrete objects (e.g., a document, contract), or people (e.g., skilled programmers). Others are capabilities for reproducing these more concrete resources on demand (e.g., the ability to hire or train skilled programmers, the ability to purchase, or write new documents).

Computer-based systems are often tightly coupled through the various elements of infrastructure, both horizontally (through a particular production lattice) and "vertically" through various layers. In the case of many computer systems, each of these layers is itself a relatively complex work organization. For example, when the DCRP staff used a laser printer in ICS to print a document, they relied upon their PC, text processor and all of its supporting arrangements. They also relied upon communication lines between their workplace and ICS, as well as the availability of ICS minicomputers. These were not always available, since communication lines might be congested towards the end of the academic quarter and the minicomputers might be "down" for preventive maintenance, file backups, or unexpected failures. Last, they depended upon the availability of the laser printer and its resources, such as toner, paper, and people skilled in fixing paper jams.

Computer-based technologies and arrangements for using them vary considerably in the scale and complexity of the production lattices which support them. The production lattices supporting a programmable calculator used by an engineer may be very small (e.g., sources of design data in the firm and vendor support for repairs). On the other hand, information systems which cut across many organizational subunits (e.g., accounting systems, inventory control systems) or which cut across many organizations are much more complicated (e.g., National Criminal Information Center which is operated by the FBI and whose main systems for "wants and warrants" and stolen cars are available to police forces nationwide).

Any user of a complex computer-based system used to support some instrumental lines of work, stands at the intersection of two work organizations. One work organization is framed around the instrumental tasks enacted by the computer user (e.g., production planning in a manufacturing firm, actuarial analyses in an insurance firm, land use analyses in an urban planning department, or processing various transactions -- payroll checks, work orders, purchase orders, travel vouchers). The second work organization is visible as he faces the computer system and other core resources (e.g., data sources) which support his computational universe: the production lattices which provide any of the computer based products in use.

For example, when the DCRP staff were revising a complex 200 item questionnaire in the Winter of 1989 and tried to print it with PPRO's laser printer, they faced two work organizations that were loosely coupled. Two DCRP research assistants and the PI worked intensively on developing the questionnaire in time for a survey. PPRO's office staff were working on other projects with equally pressing deadlines. As the survey deadline approached, one RA was frantically trying to refine the formatting and correct problems. She wanted substantial access to PPRO's postscript-compatible laser printer, because WordPerfect 5.0's screen preview was sometimes imperfect for displaying numerous boxes and lined borders on the questionnaire. PPRO's office staff were willing to allow her some casual, but limited usage.

This production lattice slices through several of the distinct layers of the computing infrastructure. The user of several different computer-based systems stands at the intersection of several different work organizations: his own instrumental activities and a production lattice for each computer-based system. These lattices may overlap, to the extent that the computer-based systems ride on shared resources, or they may be quite distinct. Discrete-entity analyses view the smooth coordination of these multiple work organizations is straightforward. For web analysts they can be problematic and will be the focus of our attention in the next subsection.

5.2 Infrastructure and Effective Computer Usage

It is common for organizations to underinvest in infrastructure, even training staff in the use of key applications. We have seen organizations, especially university departments, where clerical staff were given PCs, but not taught how to use key software. Clement (1990) reports an interesting case where some secretaries in a university administrative department did not know they were getting computers until they actually arrived - surprise!

Few workers are rewarded explicitly for learning or using computing; it is usually viewed as an adjunct to a job -- like a telephone, FAX, or photocopier. Job performance is usually evaluated in terms of the overall quality of goods and services one produces, rather than the technologies used to produce them. Computing applications vary in their complexity, and people vary in the range of features they need to use. But it can easily take a person 15-50 hours to learn about MS-DOS and the good formatting skills in a complex word processing package like Microsoft Word or WordPerfect.

Managers of professional groups often assume that many of their staff will teach themselves the necessary computing skills. And it is common for professionals to teach themselves some small envelope of skills with specific software so that they can use computing for some key tasks. Clerks have a different work ethos, since they are selected and trained to receive delegated work rather than working on self-generated activity. While some clerks have become self-starters in learning computing, clerical groups sometimes languish in computing use without specific training. In most cases, managers underestimate the time required for staff to develop adequate skills. Weaknesses in the skill levels at PPRO, for example, lead their secretarial staff to gravitate to Macs over PCs because they were self-taught. (PPRO's computing specialists preferred to maintain the minicomputer or their statistical software rather than get deeply involved with secretaries and their word processing).

The expected computing infrastructure can influence the choice of an appropriate computing package. For example, in mid-1987, I was interested in selecting a relational database for the DCRP and was also asked to recommend such a database for an office in the university administration staffed with non-technical administrative analysts. Amongst other alternatives, I evaluated Dbase III and Paradox 2.0. Dbase III was the market leader, but it required programming to develop moderately complex inquiries and to produce all but the simplest reports. In contrast, Paradox could produce complex queries and all of its reports from a menuing system, although it also had a programming language.

If the groups expected to develop complex custom applications, and had funds to retain programmers to develop and refine applications, then Dbase III would offer significant advantage in finding programmers. If the groups had to develop their own applications, and periodically devise their own reports, then Paradox would be a more effective choice. In the case of the administrative office, there were funds to buy software, but no funds to hire a database programmer. In the case of the DCRP, I did not want to devote significant funds from a very limited budget to hiring a programmer. In both cases, Paradox was the more workable alternative. This case is a simple illustration where ignoring the actual infrastructure could have lead to a poor choice. Dbase III was a market leader for many good reasons, but not good for these groups.

Decentralized computing has many costs that are not apparent to central administrators (King and Kraemer, 1981). While the purchase price of PC hardware and software can be relatively affordable, there are many hidden costs for a variety of activities associated with computer use. Many of these costs are measured in the time such as training new staff, archiving files, and upgrading and maintaining computing. They are frequently difficult to quantify precisely. With limited budget, a work group has incentives to develop their own strategies for balancing between the perceived costs of outside computer services, the procurement costs for hardware and software used within the group, and the time required to develop or improve in-group computing capabilities.

Providing resources for secretarial groups and helping them establish such a support network has been shown to help these groups become self-supporting computer users. Clement (1990) describes clerical work groups that were given such an opportunity. Administrators decided unilaterally to procure microcomputers and require secretaries to use them. Most of the funds were spent on hardware and software, and training was minimal. The secretaries were unhappy with their computing infrastructure. They had to learn computing skills in the course of their normal work and relied heavily on outside expertise.

Because of secretarial complaints to the unions and to the administration, a special project, called the Self-Managed Office Automation Project (SMOAP), was started to see if the office workers could support their own computing needs (Clement, 1990). By establishing a microcomputer resource center, which the secretaries used for instruction, problem solving and as a center for reference materials, the secretaries became self-supporting. They were able to develop a support network of clerical users that crossed departmental lines. Their evaluation of their computing support infrastructure improved dramatically. Through the help of SMOAP, each group of secretaries moved from being dependent on external expertise to developing their own expertise.

It would be no surprise if one found that people who were not trained to use a particular computer systems didn't use it. But the influence of infrastructure can be far reaching, even when it is more subtle. The case of group calendars is an interesting illustration.

Many system designers view information as a public good, one that systems users are likely to share readily. This assumption underlies the design of many computer conferencing systems, and one emerging application, group calendars. The motive for group calendars responds to professionals' common complaints about the amount of time and number of phone calls they require to schedule meetings with 3 or more participants. Certain popular commercial offices systems, such as IBM's PROFS, DEC's All-in-One, and Data General's CEO include group calendars. Some specialized systems, such as Higgins and Right Hand Man, are built around calendars as a core application.

In a review of "workgroup productivity software," Derfler (1989) argued that group scheduling or calendaring software was a critical module," although other modules, such as text processing and electronic mail, are important to make a more usable system. He went on to say:

> Scheduling three or more busy people for a meeting, along with arranging for a conference room and a slide projector, can be a frustrating and time-consuming task, requiring at least three phone calls. If one person or facility isn't available at the time the other people or facilities are, a whole series of negotiations begins. Mathematicians refer to it as progressive approximation; you (or your secretary making the arrangements) call it frustration. Before the scheduling problem is resolved, the number of people involved and phone calls made may have increased dramatically.
>
> Scheduling programs ... vary in how they confirm proposed events. The simpler packages assume that if the event fits on the calendar, that the people scheduled to attend will be there. Other programs ask for confirmation, while some go as far as to tie into electronic mail modules for notification.
>
> .... The best scheduling software is utterly useless if people aren't willing to play the game by keeping their personal calendars current. Obviously, these personal calendars are at the heart of the group scheduling process-- calendars that aren't readily available or easy to use will never be maintained by group participants. With this in mind, it seems imperative that these programs allow you to run the personal calendar module (interactively while running other programs) and make it easy to use (Derfler, 1989:248).

Derfler describes and critically evaluates key features of some major programs, and describes the best of these packages as dreams come true for busy professionals and managers. He describes how these programs can facilitate various kids of group activities, such as scheduling, under the best of conditions: machines are up and running properly; people have immediate access to the shared system to keep their calendars up-to-date; people actually keep their calendars up-to-date. Unfortunately, he does not explain what social conditions make these packages most effective -- or even usable at all.

Recent empirical organizational research on the use of groupware shows that group calendars are rarely used for scheduling meetings (Bullen and Bennett, 1991). The conventional explanation focuses on the interface designs, and critiques calendars whose use requires alot of additional work -- clumsy systems, systems which have different conventions than the professionals' primary applications, and the work effort sometimes required to leave one's primary application to jump to a calendar. These criticisms can all be resolved by improved software designs.

Grudin (1989) asks a more fundamental question: how valid is the model of information processing as a public good which is implicit in group calendars? Grudin observes that managers, who have good secretarial support, are more likely schedule meetings than are professionals who rarely have dedicated secretaries. In this common set of cases, the "political economy of effort" to keep calendars up-to-date works to the advantage of managers (low cost). But it imposes relatively high demands on professionals who do not have comparable assistance (infrastructure) to keep their calendars current. While interface improvements can reduce the costs, the way that people who are out of their office may (managers) or may not (professionals) be able to phone in calendar changes plays a more fundamental role in the information economy and subsequent workability of such systems.

5.3 Negotiations Around Production Lattices

It would be analytically attractive if we could apply discrete-entity models to help understand the kinds of infrastructural resources which are available to participants who develop, integrate, and use computer-based systems. However the way that production lattices are layered and often cut across organizational units usually makes discrete-entity analyses too weak.

The natural open systems organization models helps us understand variations in the kinds of resources which are (or could be) available to different parties. Discrete-entity models lack the conceptual richness to easily explain phenomena such as these:

a. When some computing resources are shared by several participants (e.g., individuals, organizational subunits, coalitions), service will often be differentially provided, and more powerful participants will often receive better levels of service where the provider has discretion. For example, PPRO's office staff were responsive to the director's preferences for using Microsoft Word, but were unwilling to inconvenience themselves by acquiring a word processing format conversion package to help work more effectively with research staff who found Microsoft Word to be technically unattractive.

b. Computing resources may be selected so as to improve the negotiating posture of service providers in conflicts which are defined at larger scales of action than a particular bundle of computer services. For example, we studied a multi-divisional engineering firm in which the primary division used mainframes supplied by a major computer vendor. Another division of the firm chose a different primary vendor, so they would have a plausible rationale to acquire or develop their own (incompatible) software. Otherwise, their engineers might have to accept software chosen by the primary division.

c. Resources may be selected in line with social meanings defined in larger social worlds in which some key actors participate. For example, systems programmers and facility managers may have special preferences for equipment identified with specific vendors (e.g., IBM, DEC, Apple) of technological standards (e.g., Unix, MSDOS). When performance criteria are ambiguous, social definitions may dominate decisions about adoption.

d. Sometimes there are conflicts between the kinds of computer-based service a consumer desires and what a provider sees as his line of work or the going concern of his organization. A common, relatively individualistic example is the difficulty many managers and users find in having programmers adequately document their software. Programmers receive many rewards -- both psychological and in their careers -- for writing programs. Documentation is not their line of work, and does little to advance a programmer's career on the job market.

In each of these four kinds of situations, the computing resources and the services provided are probably "sub-optimal" based on narrow considerations of task. The resources selected are based in part on negotiations which take place in social settings defined by boundaries that extend well beyond the focal computing resource and its immediate users.

Negotiations matter in practice. Technical efficiency analyses only explain a fraction of the kinds of choices organizational participants make when selecting or providing computer-based resources. Moreover, today's structural arrangements (e.g., routines for allocating computing resources) are often the byproduct of battles or negotiations worked out previously.

5.4 History of Computing-related Commitments

Over time, key actors developed an array of infrastructural resources to support their computer applications. Discrete-entity analysts usually take these resources for granted. Web analysts view existing arrangements as embodying a complex set of commitments made earlier. Some of these commitments are particular to computing (e.g., the kind of equipment in use). Others are defined in larger settings within and without the organization (e.g., hiring procedures, procurement procedures).

Few published studies of computing arrangements carefully trace the way that commitments made over long time periods influence the use of a computer application, how its constituencies shape and adapt it, etc. The studies that examine historically derived commitments usually focus on periods of a few years (Kling, 1978b; Markus, 1984; Dutton, 1981, Kling and Iacono, 1984b; Jewett and Kling, 1991) although longer periods are sometimes examined to provide "context" (Laudon, 1974; Laudon and Laudon, 1988:879-896).

Organizations develop infrastructures for computing support over months and years -- the routines, the equipment selected, the expertise developed in different subunits, the meanings and ideologies developed to justify them, etc. These are not easily altered when troubles arise. In general, infrastructures are developed incrementally, rather than by large and swift changes. Discrete entity analysts try to identify the problems of computer use by focusing on specific procedures or equipment. If one views infrastructure as a product of short-term organization building, then it should be easy to change when needed. Web analysts are usually less sanguine that infrastructures can be rapidly created or altered by simple administrative will.

Any analyst of computing developments tacitly selects a temporal frame in which to situate his account. Discrete entity analysts rarely begin their accounts before the initiation of the focal computing resource. Such historical frames are often too limited to help explain the kinds of constraints faced by key actors in the local saga of computing.

For example, consider the DCRP's replacement of PC-Write 2.7 with WordPerfect 5.0 as the staff's primary word processor in mid-1988. A discrete entity analysis would examine the corresponding features of PC-Write, WordPerfect and other alternatives, such as MS-Word and Emacs+Tex. The analysis would match their features to the kinds of documents produced by the DCRP. Such an analysis would miss key contingencies and overvalue others. The ICS faculty and support staff had a commitment to EMACS and Tex that went back over 5 years, but there was no formal training for Tex. Tex was relatively complex and the manuals were written primarily for programmers. Further, the DCRP did not have computers or software which would allow the staff to readily preview documents in Tex. The choice of Tex would have facilitated the use of ICS' Imagen laser printers, but would have imposed relatively large training and operations costs. In fact, the consideration of WordPerfect was an attempt to gain some of the laser printing advantages of Tex with lower costs of learning and use.

The PPRO staff was committed to MS-Word for over 3 years. They were also moving from PCs to Macs, partly to provide new graphics services, and partly to offset their own poor training with PCs[10]. In 1988, MS-Word was the only word processor which worked well on both Macs and PCs. PPRO's and ICS' text processing arrangements were highly institutionalized and their staffs had little incentive to change to newer software, like WordPerfect 5.x.

The DCRP staff could choose the same word processors as were used by either ICS or PPRO. The print quality standards in DCRP's the research world had shifted rapidly so that laser printing (rather than dot matrix printing) was relatively standard and a signifier of professional quality work.

Further, space arrangements at PPRO were stable since 1982, and also institutionalized. The DCRP did not have sufficient space at PPRO for the PI and RAs to have quiet space, despite continuing requests to PPRO's managers. The PI and RAs often worked at home when they were writing papers and most likely to want laser printing services. WordPerfect's ability to allow users to preview the formatting of laser printed documents, and then to send the documents over communication lines to ICS offered outstanding advantages. These outweighed the advantages of working with PPRO's standard (MS-Word). Since the PI still used ICS facilities for printing administrative and teaching materials, and the ICS RA's used ICS for printing course papers, the solution to match PPRO would not have been a strong overall solution for them. The DCRP PI's selection of WordPerfect over MS-Word, however, increased the negotiating costs between the DCRP and PPRO's administrative staff. It also increased his workload at ICS, since his secretary was trained to use Tex. But the ICS Department Chair refused to give him secretarial support for using PC packages, like WordPerfect.

6.0 LOCATING INFRASTRUCTURE IN A WEB OF SOCIAL RELATIONSHIPS

In this chapter I've illustrated the role of computing infrastructure in making systems more or less workable. I would like to conclude with a neat method so that any analyst could readily identify the infrastructural resources which are critical for a computer system and workable for her organization. This method would provide a satisfying conclusion to a long chapter. Unfortunately, I do not have a such a method. Nor is the search for a simple method a good way to learn from this chapter. The case of the DCRP and the subsequent discussion illustrates many key ideas about the way that computing infrastructures are tailored for specific systems and social settings.

It is daunting to make prescriptions that would fit all organizations, even when the prescription is a process. Organizations differ in numerous ways that shape the appropriate forms of computerization. Some are richer, other are poorer and less able to afford the staff support costs to effectively run complex systems, even when they can scrape to buy the core technologies. Some organizations are highly entrepreneurial and their managers and professionals are eager to find new ways to organize work. Other organizations are drifting, and may try to computerize an activity because their managers were told to or they are copying "what everyone else is doing." Even though the pace of corporate mergers has slowed in the US since the late 1980s, some organizations are still purchased, and forced to adopt information systems that match those of their new owners' firms. In multidivisional firms, some groups explore the leading edge of high tech in some application; other groups may later be adopting systems which have now become a corporate standard.

These are some of the different organizational conditions which frame the social stages upon which people play out their the dramas of computerization. There are also important variations in the kinds of computerized systems which are the focus of these dramas -- from text processing for a small office through a rich mix of system in an engineering lab to large-scale multi-organizational information systems. The appropriate strategy for an analyst depends upon the scale of the system and the conditions under which it is being adopted or expanded.

Unfortunately, many organizations **underinvest** in infrastructure for computerization projects (Kling, 1987; Kling and Iacono, 1989; Clement, 1990). I believe that the dominance of discrete entity analyses is one cause, although organizational budgeting practices are another. Discrete-entity analysts easily draw defining boundaries around computerized systems and their users. They can focus on a computer-based systems and a single user, a work group or some formally defined unit, like a public agency or a business firm, and their goals or key tasks. In the case of the DCRP's they would focus on the DCRP, and perhaps include it's administrative home, PPRO. While they often ignore infrastructure, they would most likely include support technologies, like communications, and training in systems use.

In contrast, web analysts have a more difficult time in defining crisp boundaries and identifying a precise computing infrastructure without examining concrete organizational arrangements. In the case of the DCRP, the boundaries include the PI's academic department, ICS, as well as the availability of students to join the project in various ways. The web analysis includes the variety of locations where DCRP participants actually work, not just their official on-campus offices. It includes other organizational networks that the DCRP is embedded within, (e.g., granting agencies who have specific rules about computer purchases; computer vendors who sometimes give equipment gifts to ICS). The temporal boundaries go back in time to the periods when key arrangements were set up.

In the extreme, a systems analyst can take one of two postures towards a client organization: (a) give them solutions to the problems they identify; (b) help raise their competence in identifying their preferences for computer systems and help them better identify what they want form systems and what resources they can devote to them. In practice, systems analysis is a mix. But the rationalistic approaches lean toward methods for delivering fixed solutions, while process approaches lean towards helping build competence as well as suggesting solutions.

I prefer the process approaches, like socio-technical design and soft systems methodology. This chapter suggests that a process oriented analyst can better help her clients by encouraging them to examine infrastructure issues early on in systems designs. But there are several ways than an analyst or group can approach infrastructure: (a) computing infrastructure will be developed along with a new system, and they do not set significant limits on the range of appropriate systems; (b) infrastructural resources are relatively rigid and an appropriate system should fit existing resources.

Our discussion of group calendars in Section 5.2 suggested that without adequate clerical support, professionals might not find the application workable. Computerization often involves changing organizational practices, policies, resources, and control patterns. "Conservative" computerization projects leave most practices and control patterns intact, and they often reinforce existing distributions of power. The development of workable computing infrastructure also entails organizational change. But even conservative computerization projects might require significant redistributions of resources and control to make them workable. The systems analyst is usually dealing with more than the immediate task and technology, since she is usually facing institutionalized practices, patterns of organizational control, and the systems of meanings in work cultures.

It is ironic to invoke institutional analyses to understand computerization in action. The images of computerization -- new technologies, innovative practices, and a patina of "revolution" -- are diametrically opposed to the stodgy images of institutionalization. The rhetorics of innovation, transformation and revolution emphasize possibilities. These rhetorics deny that historical patterns will continue to shape the future. In fact, computerization has not transformed many organizations as rapidly as some advocates hoped (Kling, 1991). Computing innovation need not fail only because of powerful organized resistance. A history of complex social and technical commitments may structure the social organization of computing and related work so as to make innovation relatively expensive and complex (Kling and Iacono, 1989). The case of group scheduling software is indicative.

Systems analysts are activists who play an important part in the design and redesign of organizations, even though many would rather see themselves as dutiful functionaries. This chapter has examined how developing computing infrastructure is an critical element in organizational designs for workable information systems.

7.0 REFERENCES

Allison, Graham (1971). *The Essence of Decision*: *Explaining the Cuban Missile Crisis.* Little, Brown, and Co., Boston, MA

Becker, Howard S. (1960). "Notes on the Concept of Commitment", *American Journal of Sociology*. *66*: 32-42.

Becker, Howard S. (1982). *Art Worlds.* Berkeley: University of California Press.

Bullen, Christine and John Bennett. (1991). "Groupware in Practice: An Interpretation of Work Experience" in Dunlop, Charles and Rob Kling (Ed.) *Computerization and Controversy: Value Conflicts and Social Choices*. Academic Press, Boston.

Checkland, Peter. (1992) Information Systems and Systems Thinking: Time to Unite? in *Challenges and Strategies for Research in Systems Development.* William Cotterman and James Senn (Ed.). John Wiley, London.

Christiansen, Ellen. (1990). "On Organizational Competence in System Development: An Activity Approach" pp 107-126 in Bjerkness, Gro and Dahlbom et. al. (Eds). (1990). *Organizational Competence in System Development*. Studentlitteratur, Lund Sweden.

Clement, Andrew. (1990). "Cooperative Support for Computer Work: A Social Perspective on Empowering End Users." *Proc 1990 Conference on Computer Supported Cooperative Work.* Los Angeles, October.

Clement, Andrew and Darrell Parsons. (1990). "Work Group Knowledge Requirements for Desktop Computing." *Proc 23rd Hawaii International Conference on Systems Science.* Kona Hawaii. Jan 2-5, (1990).

Cyert, Richard & James March (1963). *A Behavioral Theory of the Firm.* Prentice-Hall, Englewood Cliffs, NJ.

Danziger, James, William Dutton, Rob Kling, & Kenneth Kraemer (1982). *Computers and Politics: High Technology In American Local Governments.* Columbia University Press, New York.

Derfler Jr., Frank. (1989). "Imposing Efficiency: Workgroup Productivity Software" *PC Magazine.* 8(16) (Sept 26). 247-269.

Dimaggio, Paul J. and Walter W. Powell (Eds.). (in press) *The New Instiutionalism in Organizational Analysis*. Chicago: University of Chicago Press

Dunlop, Charles and Rob Kling. (1991). *Computerization and Controversy: Value Conflicts and Social Choices*. Academic Press, Boston.

Dutton, W. (1981). "The Rejection of an Innovation: The Political Environment of a Computer-based Model", *Systems, Objectives, Solutions. 1*(4) (November): 179-202.

Dutton, W.H. and Kraemer, K.L. (1984). *Modeling as Negotiating*: *The Political Dynamics of Computer Models in the Policy Process.* Norwood, N.J.: Ablex Publishing Company.

Gasser, Les. (1986). "The Integration of Computing and Routine Work." *ACM Transactions on Office Information Systems*. **4**(3)(July):205-225.

George, Joey, Suzanne Iacono and Rob Kling. (1991). "The Social Organization of Computing Skills: Expertise and Learning in Four Types of Desktop Computing Environments" Unpublished Working Paper. Department of MIS, University of Arizona, Tucson.

Goodman, S.E. (1979). "Soviet Computing and Technology Transfer: An Overview", *World Politics.* Vol. 31, pp. 539-570.

Grudin, Jonathan. (1989). "Why groupware applications fail: problems in design and evaluation." *Office: Technology and People, 4*:3:245-264.

Hirschheim, R.A. (1985). *Office Automation*: *A Social and Organizational Perspective.* John Wiley & Sons: New York City.

Iacono, S. Kling, R. (1987). Office technologies and changes in clerical work: A historical perspective. In R. Kraut (Ed.), *Technology and the transformation of white collar work.* New Jersey: Lawrence Erlbaum.

Jackall, Robert. (1988). *Moral Mazes: The World of Corporate Managers.* New York, Oxford University Press.

Jewett, Tom and Rob Kling. (1990). "The Work Group Manager's Role in Developing Computing Infrastructure. *Proceedings of the 5th Conference on Office Information Systems*, Cambridge MA, April.

Jewett, Tom and Rob Kling. (1991). "The Dynamics of Computerization Social Science Research Team: A Case Study of Infrastructure, Strategies, and Skills" *Social Science Computer Review*.

King, John L. and Kraemer, Kenneth L. (1981). "Cost as a Social Impact of Telecommunications and Other Information Technologies." In Mitchell Moss *Telecommunications and Productivity*, New York: Addison-Wesley.

King, John L. and Kenneth L. Kraemer. (1986). The Dynamics of Change in Computing Use: A Theoretical Framework." *Computing, Environment and Urban Systems*. 11(1,2):5-25.

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

Kling, Rob (1978b). "Automated Welfare Client Tracking and Service Integration: The Political Economy of Computing", *Communication ACM.* Vol. 21(6), pp. 484-493.

Kling, Rob (1978c). "Information Systems as Social Resources in Policy-Making", *Proceedings 1978 National ACM Conf.* Washington, DC.

Kling, Rob (1980). Social Analyses of Computing: Theoretical Perspectives in Recent Empirical Research. *Computing Surveys.* Vol. 12(1) pp. 61-103.

Kling, Rob (1986). "The New Wave of Academic Computing: Elements of a Social Analysis." *Outlook*, v19 (1,2) (Spring/Summer 1986):8-14.

Kling, Rob. (1987). "Defining the Boundaries of Computing Across Complex Organizations" in *Critical Issues in Information Systems Research.* Richard Boland and Rudolf Hirschheim (eds.) John Wiley and Sons.

Kling, Rob. (1991). "Computerization and Social Transformation." *Science, Technology and Human Values*. *16*.

Kling, Rob & William Dutton, (1982). The dynamics of the local computing package. in Danziger, Dutton, Kling, and Kraemer [1982].

Kling, Rob & Elihu M. Gerson (1977). "The Social Dynamics of Technical Innovation in the Computing World", *Symbolic Interaction.* Vol. 1(1), pp. 132-146.

Kling, Rob & Suzanne Iacono (1984a). "Computing as an Occasion for Social Control. *Journal of Social Issues.* Vol. 40(3), pp. 77-96.

Kling, Rob & Suzanne Iacono (1984b). The control of information systems after implementation. *Communications of the ACM*, *27*, 1218-1226.

Kling, Rob & Suzanne Iacono (1986a). "Printco." in *A Casebook for Management Information Systems.* Henry Lucas, (ed.) New York: McGraw Hill.

Kling, Rob & Suzanne Iacono. (1989a). "The Institutional Character of Computerized Information Systems" *Office: Technology and People*. *5* (1):7-28.

Kling, Rob & Suzanne Iacono. (1989b). "Desktop Computerization and the Organization of Work." in *Computers in the Human Context: Information Technology, Productivity, and People*, Tom Forester (ed.) Cambridge, Ma:The MIT Press.

Kling, Rob, Suzanne Iacono & Joey George. (1990). "Occupational Power, Patterns of Desktop Computer Use, and Quality of Worklife." in *Desktop Information Technology: Organizational Worklife in the 1990s*. Kate M. Kaiser and Hans J. Oppeland (eds.) Elesevier/North Holland. Amsterdam.

Kling, Rob & Walt Scacchi (1982). "The Web of Computing: Computing Technology as Social Organization", *Advances in Computers.* Vol. 21, Academic Press: New York.

Kraemer, Kenneth .L., Dickhoven, Siegfried, Fallows-Tierney, Susan, and King, John L. (1985). *Datawars: The Politics of Modeling in Federal Policymaking.* New York: Columbia University Press.

Kraemer, Kenneth L., Perry, James, King, John L. and Dunkle, Deborah. 1988. "Institutionalization of Technological Change in Public Organizations." PPRO Working Paper. University of California, Irvine, August.

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

Laudon, Kenneth C. and Laudon, Jane P. (1991). *Management Information Systems: A Contemporary Perspective* (2nd Ed.). Macmillan Publishing Company, New York.

Lepore, Stephen, Rob Kling, Suzanne Iacono, and Joey George. (1989) "Implementing Desktop Computing, Infrastructure, and Quality of Worklife." *Proceedings of the 10th Annual Conference on Information Systems*, Boston MA, December.

Long, Norton (1958). "The Community as an Ecology of Games" *American J. of Sociology* vol. LXIV (November): 251-261.

Markus, M. Lynne. (1984). *Systems in Organizations: Bugs and Features*. Pitman Pub. Co, MA.

McHenry, William and Seymour Goodman (1986). "MIS in Soviet Industrial Enterprises: the Limits of Reform from Above." *Communications of the ACM,* **29**(11)(November):1034-1043.

Perrow, Charles. (1986). *Complex Organizations: A Critical Essay* (3rd ed). McGraw Hill. New York.

Scott, W. Richard (1987a). *Organizations: Rational, Natural, and Open Systems* (2nd Ed.) Englewood Cliffs, NJ: Prentice Hall.

Scott, W.Richard (1987b). "The Adolescence of Institutional Theory" *Administrative Science Quarterly,* 32: 493-511.

Shulman, Arthur D. Robyn Penman, and David Sless. (1990). "Putting Information Technology in its Place: Organizational Communication and the Human Infrastructure." pp. 155-192 in John S. Carroll (Ed.) *Applied Social Psychology in Organizational Settings*. Lawrence Erlbaum, Hillsdale, NJ.

Steinbruner, John (1974). *The Cybernetic Theory of Decision*: *New Dimensions of Political Analysis.* Princeton University Press: Princeton, N.J.

Strauss, Anselm (1978). *Negotiations*: *Varieties, Contexts, Processes,* and *Social Order.* Josey-Bass Inc.: San Francisco, CA.

Walsham, G., Veronica Symons, and Tim Waema. (1988). "Information Systems as Social Systems: Implications for Developing Countries" *Information Technology for Development 3*(3).

Wood-Harper, A. Trevor and S.W. Corder. (1988). "Information Resource Management: Towards a Web Based Methodology for End User Computing." *J. Applied Systems Analysis*. *15*, pp.83-99.

This conception differs from that of Shulman, Penman and Sless (1990) who rigidly distinguish between the "technical infrastructure" in organization and the "human infrastructure. While we have many points in common with their analysis, web models hold that human and technical infrastructure are highly interdependent. Treating them in isolation is a fundamental error. It also differs from the conception of King and Kraemer (1986:Fig 7) who identify the "internal computing infrastructure" as all internal computer systems, all computing applications, their uses, and related the technical knowledge. This conception of computing infrastructure merges together all computing in an organization. It does not readily enable the analyst to separate infrastructure into hierarchy of layered systems, skills and organizational practices which support activities "above them."

[1] Analysts who argue that computers dehumanize work can also rely upon discrete-entity models (See Kling, 1987).

[2] Tom Jewett played a key role in developing this case. For a more detailed examination of this case, see Jewett and Kling (1991). In 1988 and 1989 two student teams also studied computerization and the work of the DCRP through interviews with project staff and their own observations.

[3] The DCRP PI finally acquired such a package in 1988, but he found that the automatic translations were imperfect and required significant time to clean up minor formatting problems.

[4] Tex is a cumbersome language in which many formatting commands are spelled out into a manuscript (e.g., *\centerline{line of text}* to center a line, *\vskip .2in* to create some blank lines); certain key punctuation, such as $ signs, must be placed between special characters because they also act as Tex commands.

[5] The PI considered acquiring a PC based version of Tex, but it was relatively expensive, slow and comparably cumbersome.

[6] Microsoft Word 4.0 allowed a user to change the background color of the text screen, but not to change the foreground color from dull grey. The PI prefers to edit with a palette of bright white text on a blue background, which reduces strain on his eyes. This version of MSWord increased is eye strain when compared with either PC-Write or WordPerfect which could be configured for the color palette which he prefers. In addition, one could not edit the file names and pathnames of files when managing files in MSWord. Since the DCRP uses numerous specialized subdirectories to help organize files, the clumsiness of manipulating file designations was also an added burden. In contrast, both PC Write 3.x and WordPerfect 5.0 allow users to create a new filename and path by editing an existing one. While MSWord had many attractive features, the PI felt that these features were very unattractive.

[7] This usage differs from King and Kraemer (1986:Fig 7), where they characterize "internal organizational context" in terms of social relationships, social structures, attitudes (such as dispositions to use technology), and the distribution of skills (such as knowledge about how to use technology). Analysts use the term "context" in a wide variety of ways. It's important to understand which elements of social life and systems a particular analyst denotes by the term "context."

[8] We are not optimistic about the prospects of creating good dynamic theories of computerization since organization-level theories of structural/political change and stasis are not well developed either. Even so, this is terrain opened up by changing technologies and technologies of change.

[9] See also the Printco case which illustrates how critical infrastructure can be in developing workable information systems (Kling, 1987; Kling and Iacono, 1989)

[10] The PPRO staff often edited working papers and reports which had 84 character lines (12cpi). The office staff complained to us that "PCs" didn't allow them to see a whole text line on their (80 character wide) screens. Even though they could have changed the fonts or margins in MS-Word so they could view complete text lines on-screen, the PPRO director did not help insure that his office staff was trained for such basic skills. However, the office staff had acquired some Macs for graphics, and found that they could more readily change fonts to see complete text lines on the screen. They found the Mac version of MS-Word easier to teach themselves and use, and effectively lobbied for a shift from PCs to Macs.