Title
The dynamics of computerization in a social science research team : a case study of infrastructure, strategies, and skills

Authors
Jewett, Tom
Kling, Rob

Publication Date
1990

Peer reviewed

The Dynamics of Computerization In a Social Science Research Team: A Case Study of Infrastructure, Strategies, and Skills

Tom Jewett
Rob Kling

Technical Report 92–43

Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717
kling@ics.uci.edu
(714) 856-5955
December, 1990

ABSTRACT

This paper examines the dynamics of Computerization in a PC-oriented research group through a case study. The time and skill in integrating computing into the labor processes of research are often significant "hidden costs" of computerization. Computing infrastructure plays a key role in reducing these costs may be enhanced by careful organization. We illustrate computerization strategies that we have found to be productive and unproductive. Appropriate computerization strategies depend as much on the structuring of resources and interests in the larger social setting, as on a technical characterization of tasks.

From: Social Science Computer Review 9(2)(Summer 1991):246-275.

INTRODUCTION

This paper examines the skill issues and hidden time demands of vertically integrating computing into a scientifically oriented research work group in a university setting. It is based on participant observation by the authors during a period of several years[1]. The research group's primary substantive focus has been to understand the way that computerization alters worklife -- based on surveys and observational studies in organizations. But we have found that our own experiences in computerizing parallel those of many organizations we have studied.

The examples for this article are drawn from the experiences of a social science research team and should have special resonance for other social science researchers. Where we have fallen into traps, perhaps we can mark them for others to avoid. Where we have developed techniques that seem to work for us, others may be able to apply them even more effectively. This paper may help social scientists better understand how social processes shape the computerization of research work and how computerization influences the subsequent content and organization of research work. As in most case studies, the detailed social conditions are not necessarily commonplace. The virtue of the case is that it can illuminate complex and common social dynamics of computerization.

In the following sections we will introduce the concept of computing infrastructure, describe the research project which is the focus of this case study, and describe the changing computer technologies used by the research team. We then examine strategies for obtaining computer-based services based on subcontracting or integrating them into the work group. In the main section, we examine the dynamics of computerization through numerous episodes. These illustrate common dilemmas and choices for managing research computing which involve hardware, software, and staffing. Since our computerization strategies are socially situated, we devote the first sections of the paper to describing important contextual elements of the research project.

COMPUTING COMPLEXITY AND INFRASTRUCTURE

Research computing does not need to be complex. A minimal computing environment might provide only word processing and data management — and the latter task might be contracted to a service agency outside the research group. But many researchers will prefer more computer support, such as statistical analysis, graphics, and electronic communication with colleagues. As researchers integrate more computing within their own groups, their computing environments become increasingly complex. How these choices are made, and how they affect the work life of a research group, are key themes which we will examine in this paper.

Much of the literature about computerization in workplaces is imbued with an optimistic "more is better" perspective which focuses on equipment (cf, Giuliano, 1982; Poppel, 1982). We characterize this perspective as "technological utopianism" (Kling, 1990; Dunlop and Kling, in press). Even in careful review journals, there is frequently an implicit assumption that selecting the "right" software and hardware are the key choices for successful computerization.

More complex computing is not just a set of expensive acquisitions. In their report on social science computing Anderson and Brent (1989) warn that "a most serious problem is that of having sufficient human resources — consultants, assistants, and others — with the expertise and time required to select, install, implement, and operate the required software (and hardware) successfully." They have succinctly identified a key point, which is often glossed, and which we expand in this paper: computing support is often a "hidden cost" and underestimated. *Infrastructure* is a useful concept in analyses of computing support — it denotes all the resources and practices required to help people adequately carry out their work (Kling, 1987; Kling and Scacchi, 1982). Infrastructure for computing refers to a variety of organizational arrangements for supporting computing, including recharge systems and purchasing procedures, as well as the human resources which Anderson and Brent identified. As the computing environment becomes more complex, so must the infrastructure to support it.

Our own research has focused on infrastructure — both for computing and for other work. We have found that work group managers influence the quality-of-work-life of participants in computerization projects by the way that they organize the computing infrastructure (Jewett and Kling, 1990). In a research group, the "manager" is the Principal Investigator (PI). And in this case study, we will examine infrastructure-building strategies that both the PI and research team members employed to support computing.

THE DESKTOP COMPUTING RESEARCH PROJECT

The desktop computing research project is the subject of this study. But it's substantive focus includes themes of this paper. It is a longitudinal study of computerization and changes in work life which was initially funded in July, 1985. We focus on work groups who have immediate access to computing "on their desktops," whether through stand-alone microprocessors, networked workstations, or mainframe terminals; we examine interactions within and between work groups; and we emphasize changes over time. We collected most of our data with a questionnaire instrument (closed-response, 200 items, with over 300 respondents in 7 major organizations) and with individual interviews of selected work group members and managers annually in 1988, 1989, and 1990[2].

Project Team Structure

The size of this research team at UCI has varied between 3 and 8 people during the four years of this study. It consists of the PI, between one and four Research Associates (a mix of PhD students and full-time staff), one or two paid undergraduates who assist part-time in data management and analysis, and from one to three undergraduates each term who undertake specific projects for academic credit[3]. In addition to the staff at UC-Irvine, we collaborate with colleagues at two other universities in the U.S. and one colleague in Belgium. They independently administer our survey instrument to additional work groups, share the resulting data with us (as we do with them), and develop research findings based either on their own samples or on the combined set of data.

One important social characteristic in university-based research teams is the wide variation in turnover rates. The PI developed the present project along with one full time research associate who remained on the project for the first three years. Graduate RAs tend to remain on the staff several years at a time, depending on their progress toward degrees or their outside career goals. The paid undergraduate staff may work for up to a year, seldom longer; many have worked for just one or two quarters. The independent-study students come and go term-by-term; a few remain for two or even three terms. Colleagues at other universities joined the project one to two years after it began. It takes a significant amount of time for new participants to learn the computing environment, as well as their substantive roles.

The other key infrastructural feature of this research team is the variation in computing interest, experience, and expertise of the individual members. The PI is a Professor of Information and Computer Science; one current graduate RA holds the MS in Electrical and Computer Engineering and another the BS in Computer Science. Other graduate RAs have come from social science backgrounds: psychology, management, or sociology. Our undergraduates generally come from the Department of Information and Computer Science (ICS), although we hire occasionally from other disciplines. We have found, however, that even ICS students have rarely developed the skills in specific operating systems, word processing, data base, statistical packages, and file management that we prefer for this project. The most computer-proficient have generally learned their craft through personal work on home computers or through off-campus employment using applications similar to ours. 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 our research team.

The overall computing expertise within our group is at least equal to that of most research (or business) work groups[4]. 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 our statistical package. Some of our undergraduates have had specialized hardware maintenance experience. But in a small group, we do not have the luxury of specialization: our members use most of the applications software to accomplish their tasks. Some staff members have had to learn virtually all of our software after joining the group. 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.

Physical Environment

Although the size of our research team has varied, our office resources have not. The project is administered by a campus social science research institute -- Public Policy Research Organization (PPRO). PPRO provides its research projects with office space, and recharges for for grant management, statistical computing and secretarial work. The project staff shares a large partitioned office with a number of other academic research groups. The project's own small (160 sq. ft.) office is delineated by modular, moveable partitions. We have two areas that face each other across an internal corridor. Each half of this arrangement is configured for two or three 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: these include 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.

The structural arrangement of the office has largely determined the structural conditions of work life in our group. 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, the graduate RAs also do much of their analytical work at home. The PI has a departmental office (in another building), but no private space in the research area. Although he could use some of our desk space, he has ceded this to the RAs and often works at home. The PI has set aside 2.5 rooms, totalling 300 sq. ft. for office space, which he shares with occasional office assistants.

Given the small research office and our unpredictable academic schedules, coordination among team members requires flexibility and ingenuity. We use electronic mail daily; we discuss issues via telephone; we frequently share files through a Sequent mainframe computer; and we meet at least once a week in a project-wide staff meeting. Managing this environment is an important element of our computing strategy.

Resources

Our research is funded primarily by grants from the National Science Foundation, by a much smaller grant from the campus, and by computing equipment from the PI's academic department. From our grants, we budget for the personnel, supplies, administration, travel, services, computing software and computer operations that are needed to conduct research and to disseminate research results.

Like many other academic research groups, our activities are also related to an academic department. The PI holds his primary faculty appointment in the ICS department and we obtained most of our PC hardware through departmental funding. However, we compete with other departmental research groups for allocation of hardware (through a committee process), and we have negligible control over the mix of resources procured by the department or the schedule on which they become available. We have benefitted form the technological tastes of ICS faculty. Most of them value engineering workstations or Macs, and we have had little competition when we sought PCs which the department was allocating to faculty and staff in 1989. But we have 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 is relatively expensive and does not interface easily with our PCs.

The PPRO office can provide some computing services (e.g., statistics on a dedicated minicomputer and word processing on PCs) but not others (such as database support for field contacts and abstracts). These services would be recharged to our grant, and would also require us to adopt specific computer file formats to match their systems.

Finally, some of our computer resources have been privately purchased by individual staff members (e.g., PCs at the PI's and one RA's home that are used almost exclusively for research project tasks). Our 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, we have much greater discretion in smaller purchases (e.g., a software package) than in major ones (such as hardware selection).

Data Complexity

The complexity of a research computing environment can be driven by the data required to conduct a specific project. Our data environment has become increasingly complex as the multi-year project has progressed. In 1985 we started with no data that were specific to the project. During the course of the project we 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 single file containing the questionnaire responses of 300 individuals, to a questionnaire with about 200 items[5]. For some research designs, this would have been sufficient. But to analyze the data by work group (a key unit of analysis)(Kling and Iacono, 1989a; Lepore, Kling, Iacono, and George, 1989) rather than by individual respondent, we created a second file which consolidated and averaged the individual responses; it included summary indices of key measures of computerization and work life. In the second and third years (1989 and 1990) we added similar data sets, for a total of six master files to be maintained, compared, and analyzed for changes over time. In addition, we have gradually added questionnaire responses from our colleagues' work groups — doubling the original files in size.

Our site contact data base, too, has grown over the years. Presently, we maintain information about organizations, work groups, key individuals, and survey details in over 12 separate database files — each contains from 10 to 20 separate data fields and from 50 to 200 individual records. Other project documents — papers, memos, reports, and correspondence — have multiplied similarly. The complexity of our data environment is reflected in our computing hardware and software.

DEVELOPMENTAL TRAJECTORY OF COMPUTING

Computerization projects are often dynamic. Their participants change equipment, the social organization of access to equipment, data, and computing skills over time. We use the term *developmental trajectory* to denote "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 our project staff changed over time in ways that we will describe below. Moreover, the project's staff participate in several academic units whose computing arrangements were also being changed along independent trajectories. Some key strategies of the projects staff were negotiated in terms of computing developments *on the campus, in other academic units, and in PPRO* (Strauss, 1978). While we focus on the research team, we will try to sketch some of these other changes so that readers can appreciate the ways that computerization projects develop within a web of extended social relations beyond the focal group (Kling, 1987).

Computing Hardware

Our computing environment has evolved substantially since the project started. Like many other research groups, we now primarily use IBM-PC clone microcomputers[6] (Treibwasser, 1987). But we do not propose to recommend specific hardware or software. Instead, we want to focus on conceptual issues that apply to most research organizations — even if they use a different family of equipment, or if their computing is mainframe-based. We will discuss our own microcomputing resources and software in "generic" terms to emphasize this focus.

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 operated 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 PC-XT clones with 20MB hard disks, dot-matrix printers, and key software[7]. 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. We found their speed seductive, even for routine tasks like word processing. Because computer science faculty preferred engineering workstations or Mac-IIs, we were able to obtain 8 PC/AT clones with little contentiousness in 1989. We have gradually replaced individual computers —- sometimes one at a time and sometimes in groups. We now have machines all of which are PC-AT style (one with a "-386" processor). A year ago, we modified some of the "-286" machines with additional disk drives that were "cannibalized" from the XT clones. Within the past few months, we have again increased the disk storage capacity of several of our computers by swapping for other PCs owned by ICS. At the same time have upgraded the video displays of a few machines to the "VGA" standard. We are more fortunate than some research groups in our access to computing equipment. Even in the cramped office, each graduate RA is allocated a PC; additional machines are shared for statistical analysis, laser printing, and undergraduate use. Because of our work arrangements, it is also important for the PI and RAs to have computers at home. The project has provided some of these; some staff members choose to use personally-owned machines.

Our printers have evolved in parallel with the computers. In the early 1970s, the PI used upper case line printers that were available on the DEC PDP10 mainframe and later the laser printers on ICS' minicomputers. For PC printing, we first used dot-matrix printers. They were adequate for statistical analysis, memos, draft manuscripts, but not good enough for final output destined for colleagues, conferences, and journals. The PI and his RAs often wrote final drafts of papers in a complex formatter (Tex) which drove laser printers at ICS. In 1988, we added two laser printers (one in the office and one at the PI's home). The early absence of a laser printer in the work group and also its later acquisition had many repercussions for our computing infrastructure.

We have used ICS' Sequent mainframe extensively for electronic mail and file-sharing. Several of the PCs in our office are direct-wired to a data switch that connects to the campus data network; with modems, we can reach the data switch from home via telephone lines. Our sharing files via a time-shared computer increases the complexity of our computing environment, since specialized techniques are required to transmit documents in the variety of formats used by PC and mainframe programs.

The project's computing environment developed from one which was based on the use of shared minicomputers for text processing owned by an outside agency (ICS) to one in which the vast majority of computer support is conducted within the work group on PCs. This shift was one choice, among several possible choices, since the group could have remained more dependent on outside computing support. Further, the PI worked to upgrade the groups internal collection of computing equipment from PC-XT clones to better equipped 286-based machines and from dot matrix printers to laser printers.

Computing Software

Our software suite has also evolved — usually in the direction of increasing complexity. During the project, we have increased the number and sophistication of our programs, and usually purchase updates to keep current with latest versions. On specific occasions, the PI has initiated a change in key software (e.g., word processors) or supported certain changes initiated by RAs.

The PI and his students began using the text processors, formatters, and mail services that were available a university-owned DEC PDP-10 mainframe in the 1970s. They shifted to similar programs that ran on his departments' DEC PDP20 minicomputers in the early 1980s. When he purchased a PC with his own funds in 1983, he experimented with a spreadsheet, two word processors, several databases, a project scheduler, a communications program, and numerous utilities. However, he and his primary research associate continued to use a minicomputer provided ICS for much of their writing. When the project began in 1985 the PI and his research associate were committed to certain minicomputer and PC software.

The PI and some of the RAs who came from ICS have comtinually read about developments in PC technologies -- both to follow the world of desktop computing for the substance of the research project and also as an avocation. This may have given the project team a bias towards developing a high performance computing environment, but one which was limited by significant financial constraints.

On the PCs, we first used a fast and high quality text processing program, but soon supplemented it with a slow and cumbersome "add-on" package for "laser-like" visual quality from the dot-matrix printers. In 1988, we converted to a new word processor at the cost of re-training the staff and re-formatting many papers and reports. We retain our original word processing program for ASCII text editing such as editing system files. In addition, we have used three different "outliner" programs to organize interviews, develop meeting agendas, and to brainstorm papers.

Talks about our survey data can be enhanced with graphics, to display group differences or temporal patterns. Since the PI was impressed with graphics used by colleagues in conference talks, he wanted to be comparably lucid. In addition, he hoped to use graphics to simplify exploratory data analysis. We began with one leading graphics package, but later converted to another package that was much easier for new staff to learn and use casually. Neither of these programs are fully compatible with our word-processing or data base software, despite the fact it imports and exports several file formats.

Throughout the project, we have relied on a powerful and versatile PC-based statistical package. We made a strategic choice not to use the PC version of mainframe packages because they required more dedicated hardware than we were originally willing to allocate to this one activity. One cost of this choice is that our RAs have to learn a new statistical package. While it is fast and versatile, it does not have a simple user interface or convenient data sorting and report generation. Nor is it fully compatible with any of our word processing or graphics software. To bridge that gap, we employ other data base and spreadsheet programs — each with its own specific uses, advantages, and disadvantages. By careful use of import/export capabilities, we can share data between these programs, although in some cases (e.g., graphics) we still find it more convenient to transfer data manually. We also use the data base management program to organize administrative information.

Finally, we use numerous utility programs to fine-tune our computing equipment. These include a "menu" to provide ready access to applications programs, a "print spooler" for the dot matrix printers, a file manager to expedite work with tree-structured disk directories, "memory-resident" programs that enhance keyboard input, communications programs for use with ICS' Sequent multiprocessor mainframe computer, file compression programs that reduce storage space and electronic transmission time, programming languages that we occasionally use for applications development, programs to optimize the arrangement of files stored on disk and to protect the machines against "virus" infestation, and basic utilities required for the initial configuration of each machine.

Between 1985 and 1990, we have moved from a single word processor, a spreadsheet, a statistical package, a communications package, a text database, a project scheduler, and a variety of utilities to two word processors (each upgraded several times), a different spreadsheet, an upgraded statistical package, a different communications package, an outliner, a charting program, three database management programs (two flat file text oriented, one structured relational), a new project scheduler, a much wider variety of utilities (most upgraded several times), and later versions of MS-DOS.

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).

Subcontracting Versus Integrating Computing Services

Computerization strategies differ along many dimensions. A key dimension which we examine in this paper is the extent to which a group controls its own equipment and carries out its own computer work. At one extreme, a research group may contract outside the group for all computer services, leaving only original writing and perhaps electronic mail for research team members. This choice may entail less bother with computing equipment, less computing expertise in the group, and minimal infrastructure. On the other hand, it may cost increased time to coordinate with the outside agencies — and there may be incompatibilities between the researchers and their support systems. Some desired services may not be readily available from outside sources, and others may be prohibitively expensive. In this option, the focus is shifted from developing computing systems within the group to carefully using and auditing external services.

At the other extreme is complete vertical integration; that is, all computing tasks performed within the group. This option may give the researchers great control over their finished products, flexibility in work life (such as the choice of home or office work), substantial time savings in developing analyses and documents, and lower operating costs. The risks of this approach include possible requirements for capital expenditure; the cost of time for training, equipment setup, and maintenance; and the specialized expertise required. Most importantly, computing tasks may detract from staff members' concentration on research substance. There are probably few research groups with the resources to handle every computer-related task (for example, hardware maintenance at the component level).

Many research groups contracted out their computing work in the era of mainframe computing. Now that personal computing equipment has become both economical and powerful, an integration strategy is practical. Many social science research groups now lean toward the "in-group" end of the continuum, as we do. Of course, subcontracting is a rational strategy under many circumstances, such as when the group cannot afford to acquire the necessary computing equipment, does not want to develop the necessary expertise or spend the time to bother with it.

Devising a Computerization Strategy

The position of a research group along the continuum that we have described may be dictated by internal requirements (for example, the structure of a research project), external constraints such as funding, or availability of outside services[8]. The specific goals of research groups as they develop or enhance their computing environments may be incompatible or even mutually exclusive.

First, cost containment is a key concern of most research managers. 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 come from different sources; one may be more readily available than the other. In our own case, the operating budget has been the limiting factor. We have found in-group computing to be more consistent with our budgets 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. We have found it difficult to get accurate cost estimates for a variety of tasks from our local service providers. Further, in a recharge system with delayed reports, some costs can mount unexpectedly. For example, students 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, we have continued to develop the group's computing at the cost of some staff time and infrastructural complexity.

Second our desires for increased technological capabilities conflict with our preferences for containing costs. A research group may find that its research data files have grown beyond the capacity of their present disk storage media — or even beyond the capability of some software to process them. Some analyses may take an extraordinary amount of time to run on the available computers, or may be accomplished only with more sophisticated software. Computer vendors rush to offer faster processors, improved color displays, higher-quality printers, more sophisticated communication devices, and improved software. These products are marketed seductively and we sometimes have had to resist finding "requirements" for them in the work group. But as we will illustrate from our own experiences, the actual cost of a computer upgrade — in time, effort, and disruption — is likely to far exceed the purchase price.

A third, and sometimes overlooked, value is ease of use and maintenance — for both hardware and software. While this value may be consistent with lower support costs, it may conflict with technological evolution. For us, 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."

A fourth important preference for our group has been to accommodate the geographical dispersion of team members between PPRO, ICS and their home offices. Not only has this arrangement set structural conditions on work life, as we discussed earlier, but it has strongly influenced our selection of computer tools to support our work. We will illustrate this point with a specific example when we discuss document preparation in the following section.

A fifth preference has been to make our computing arrangements congruent with those of the organizational units with which we 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 we 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. It would not make sociological sense for other research teams to mimic these details; but the influence of economic, technological and human resources on forming strategies and implementing them, as well as the compromises inherent in all such strategies should be instructive to many other research teams.

THE DYNAMICS OF DESKTOP COMPUTERIZATION

Computerization is a social and technical process for organizing computing equipment, access to it, support for it, etc. (Kling and Iacono, 1989b). The common reference frames for examining computerization focus on the technologies as an information processing resources which can support organizational strategies through their functional roles. In this approach, the major activities in a computerization strategy are relatively large scale moves to adopt and implement major computerized systems and to set organizational practices (cf. Ward, Griffiths and Whitmore, 1990; Walton, 1990). The dynamics of computing this "systems rationalist" perspective focuses on major organizational changes, but treats the abilities of organizational elites to implement their preferences as relatively unproblematic. Our approach is anchored in a view of organizations as organized coalitions bargaining in overlapping contexts and playing different "games (Strauss, 1978; Kling, 1987)." Our dynamic analyses examines the problematics of computing implementations negotiated in a web of social relationships and technological contingencies (cf. Gasser, 1986; Kling and Iacono, 1989b; Kraemer, King, Dunkle, and Lane, 1989; Walton, 1990). It examines the technical and organizational feasibility of tactics as well as strategy by examining the ways how actors develop them while negotiating between multiple, possibly conflicting commitments -- such as between different organizations or between teaching and research (Kling 1986).

Computing strategies which may appear "rational" for an organization (or work group), may be infeasible because of the dilemmas of embedding the innovations in everyday labor processes. Conversely, work groups may adopt practices, in part, because of their workability. In this section we examine the dynamics of computerization in the research group by illustrating the conditions which influenced many of the social and technical choices -- as well as their repercussions.

We encountered many types of computing experiences during this project. Some pertain to tasks that are in the foreground of our research: document production and formatting, statistical analysis, data base maintenance, and electronic mail. The computing environment influenced staff selection and training. As the project progressed, we sought some hardware and software upgrades. And we found that we spent far more energy on ongoing maintenance than we expected.

We will illustrate each of these areas with episodes from the project. In each case, we will emphasize the criteria that influenced our choices, the choices that we made, and their repercussions (good and bad). These analyses are much more organized than were the discussions and perceptions at the time of these actions. Our computing arrangements are one solution to a set of project contingencies and the structuring of resources in our university.

Foreground Activities in Research Computing

We first examine the computer applications devoted to some of the overt work on the research project: writing memos and papers, managing data and communicating with team members. In subsequent sections, we'll examine the repercussions of these activities on staff skills and computing support.

Document writing, formatting, and production

A central focus of our computer use is preparing documents such as letters, memos, field notes, scholarly articles, and questionnaires. Between 1985 and 1990, we have made substantial changes in our capabilities for producing documents. Between 1985 and 1988, we drafted and edited documents on our PCs. We had three major strategies for producing laser quality papers for professional distribution. One strategy was to subcontract the final stages of editing and printing to PPRO's office staff. We gave our text files (on floppy disk) which they translated into their word processors, and reformatted for them. PPRO administrators continually encouraged us to use PPRO's staff for office services. But we found continual annoying problems. Since our machines and those of PPRO's text-processing group had incompatible word processor formats, we faced costly and tedious reformatting if we wanted to "take a document back" after passing it to them or we were faced with having the PPRO staff recharge us for all subsequent revisions. This was a particular problem with documents which we revised in several discrete rounds of review, such as journal articles. In addition, we sometimes found turnaround time to be excessive; and new errors often crept into revisions, requiring additional auditing. But the most important problem, in our view, was the high overall cost of the production services. Our interest in cost containment led us to seek alternative ways to produce complex documents, such as journal articles and questionnaires.

A second strategy was to print on a 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 customizeable. But they also require significant time to learn and a taste for programming; and documents formatted with them can be easily distorted with a misplaced character. Our senior research associate at the time was reluctant to spend time mastering Tex. Because she wrote at home, she had no consultant readily available to help with Tex's idiosyncracies. 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 the on-campus printer (to pick up output) and the office or home (to fix problems in the output) were required. Our third strategy was to use a PC-based program that significantly enhanced the visual quality of the dot-matrix printouts. It worked with our existing text editors, and could be used at any of our locations. However, it was painfully slow for long documents, since it could take well over three minutes to print each page.

In 1987, the PI directed a team of students in exploring the possibility of shifting to one of two major word processors, one of them which was then used by PPRO's office staff. We did not see adequate advantages to either one at the time. In 1988, one of these word processors was enhanced with new features which allowed formatted documents to be previewed on screen at any of our locations, and to generate Postscript files could drive ICS' laser printers without resorting to a complex command-driven formatter. While this word processor was not used by PPRO's office staff, the advantages of previewing and simplified laser printing at ICS were major incentives for changing. The PI also found PPRO's word processor to be clumsier for some key operations, even though it was a powerful and popular package.

The combination of our new word processor (with preview capability) and new Postscript compatibility for ICS laser printer enabled us to significantly reduce our trips to ICS to preview laser printed documents. But we could still not eliminate all trips to ICS. We experimented with using a laser printer used by PPRO's office staff which accepted Postscript files. They wanted us to give them files to print and to recharge both labor time and equipment use. We wanted to use it as a walk-up self-service with charges for equipment use, akin to the photocopier. Moreover, we wanted to be able to use our wordprocessor on their PC to make quick changes to documents if there were unexpected formatting problems. However, PPRO's office staff had the printer located inside their carrels and attached to PCs which they used for ongoing work. Consequently, they found our attempts at self-service to be intrusive and we were barred from continuing.

In the summer of 1989, the PI negotiated with the ICS department for two laser printers (in exchange for a complex administrative assignment)[9]. During our first year with the laser printers, we became seduced by strategies to improve their usefulness. One of our undergraduate assistants spent most of a quarter (perhaps 60 hours) testing methods for incorporating graphics-program output into word-processor documents. All were so slow, awkward to use, or lacking in quality that we reverted to traditional "cut and paste" methods to generate a report that was due.

When we used ICS' laser printer with Postscript output, the PI became used to the flexibility of arbitrary font sizes (e.g., 11.2 point). Some undergraduate students and one of the graduate research associates spent substantial time working to improve our ability to have a larger selection of font sizes than are provided by cartridges through the use of soft fonts.

Even in day-to-day writing, we are tempted to 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 we 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 can take several hours more than we might have planned.

Increasing our computer capabilities gave us a great deal of control over the final appearance of our documents, and enabled us to meet page limits and format specifications for conference and journal submissions quickly with minimal marginal expenditures[10].

Data Management and Customized Applications

The evolution of our data-entry and data management procedures is instructive. In the case of survey data entry, we faced the classic problem of insuring that the data in a computer accurately reflects the source documents (ie., our questionnaires). In all three years of the survey, we have used double-entry to enhance data integrity — but we refined data validation practices over time.

The PI considered purchasing a specialized survey data entry program, but felt that they were prohibitively expensive for a once-a-year task. Instead, we used our spreadsheet program for data entry. In the first year, one of our undergraduate assistants wrote a program (in BASIC) to compare data from two spreadsheet files, before converting it to the statistical package format. In the second year, we made minor changes to our questionnaire. The data-verification program was no longer useable without corresponding changes. A graduate research associate who preferred Pascal developed a new program to do the same task somewhat more rapidly. In the third year, we were fortunate to have an undergraduate student who had previously done data entry and was skilled in data base development. He built a data-entry system that was based on our structured database package rather than the spreadsheet, and featured easy-to-use data entry screens. Our data-entry staff then spent at least 20 hours further refining even the new, improved procedure. But this procedure was much more streamlined and less error prone than our earlier approaches.

We also developed several custom applications with our structured relational database which we selected in 1987. It has a relatively intuitive interface for non-expert users, and requires no special programming for basic data-entry, search and reporting[11]. For example, we maintain records about our research sites, such as the names and phone numbers of our contacts, the dates when we have administered surveys. We prefer these administrative data files to be rapidly accessible, complete, accurate and easy to use. This appears to be a very straight-forward task — well suited for a PC.

Sometimes even a simple task can be elusive. Organizing the data would have been easy if each of the work groups in our study were identical and structurally stable. Real work groups, however, do not fit readily into the rows and columns of a researcher's data base. They organize, re-organize, consolidate, split, move about the organization, and disband. Our points of contact may be outside the work group (for example, a higher-level supervisor), or we may have multiple points of contact within a single work group. We initially attempted to fit this data into a single "flat-file" format, but the resulting structure was impossibly baroque. Later, we developed multiple files that were linked through the "relational" capability of the data base program. Each file is now relatively easy to access and update, but the total system remains as baroque as ever[12].

We learned that there is often a "better way" to computerize a task — but that implementing it may require a continual commitment of staff time. We concluded that custom-developed software should be a "last resort" for computerized work groups, limited to those applications that simply cannot be accommodated by any other means.

Electronic communications

We use electronic mail (email) to communicate within the group, with colleagues at UCI, with collaborators at other universities, and with colleagues internationally. Email is the least problematic of our foreground tasks —- possibly because the shared hardware and software are provided by the ICS Department. We cannot modify this system, so our primary cost is a service fee recharged to our research grant by ICS. We have casually considered setting up our own bulliten board for sharing files and electronic mail, biut it would be far too costly, complex, and still require access to ICS system for mail outside the group. We are essentially "resigned" to ICS mail software — which is adequate for the task but which has the terse and cryptic command structure common to many Unix programs. Electronic communication (including mail and file sharing) facilitates the geographical dispersion and asynchronous schedules of our team: unpaid undergraduate students who are in the office periodically, the local staff who work at home, and the colleagues who work in other cities. Email has proven to be a valuable — even essential — tool[13]. There are rare occasions when ICS' mainframe "goes down" for maintenance or upgrade, or the email system slows due to over-use. We work around these problems like everyone else who uses the system.

More subtly, there is the seduction of the many local and nationally distributed electronic "bulletin boards" that are available through the email system (Dunlop and Kling, in press). The ICS department conducts a significant fraction of administrative business by email; and there are special electronic boards for general announcements, colloquia, graduate student affairs and certain courses, as well as boards devoted to a wide variety of computer science topics, social interests, and recreational topics. Reading even a small selection can be time-consuming, and much of the material is essentially "junk mail." We rely on the ICS students paid by the project to budget their time so they don't spend inordinate time managing their email and scanning bulletin boards.

Conclusions about foreground activities

The PI's interest in having high performance computing support for major tasks such as writing, data management and statistical analysis *at modest direct cost* "drove" key choices about computerization. Our computing arrangements are one solution to a set of project contingencies and the structuring of resources in our university. Some preferences were non-rational in that they were based on behavior that was becoming institutionalized in our research world (Kling and Iacono, 1988;Kling and Iacono, 1989b). We adopted institutionalized standards, such as presentation graphics and laser printing, as soon as they were clearly becoming academic world conventions. The criteria for selecting specific equipment to implement these choices were varying mixes of the the five criteria discussed above. We were not exempt from the short term opportunism of many decision-making processes. We took advantage of "target of opportunity" equipment and personnel availability; we reacted to "crises of the moment" (such as publication deadlines); and we frequently improvised solutions to computing problems as priorities and personnel changed over time.

We have also learned that there are many hidden costs associated with computerization — most importantly, staff time. The strategy of developing significant computer capabilities within the group led the PI to recruit ICS undergraduates with good computing skills. They played a critical role in supporting our computing equipment, but they also had a bias to experiment with equipment and also stimulated some of the project's technological dynamism. We will further explore the nature of staff skills in the next section.

Staff skills and training

All of the students who work on the project arrive with some sort of computing skills. Many of the graduate students and undergraduates own PCs. Even so, when they join the project, they usually do not know our suite of application software and usually they don' have the sophistication of to manage the complex array of project files.

Software skills

Imagine the following advertisement: "Immediate job opening: a nationally-recognized research group is seeking candidates to fill half-time appointments as research associates. Applicants must hold a graduate degree in sociology or a related field, or in computer science with an emphasis on social aspects of computerization. Skills required include publication-quality writing, knowledge of the research literature, ability to analyze survey data, and experience in field survey techniques. Required computing skills include familiarity with personal computer and mainframe operating systems; experience with computerized text processing, statistical analysis, data base management, communications and file management; and the ability to trouble-shoot and resolve problems with computing software and hardware. Must be willing to re-locate to an expensive area at an entry-level salary."

This combination of qualifications for an ideal RA is unrealistic, since graduate students come to learn research skills. We teach research substance and process in the course of the work. But we also teach a significant body of computer skills Even our undergraduates — upper-division computer science majors — often require extensive training in our specific software. They also require training in practical procedures to effectively manage software and data in a group setting -- such as carefully labelling diskettes and informing others about ways that they reconfigure PCs[14].

Even though we study computing infrastructure, we have not developed a highly sophisticated infrastructure for own organization[15]. Our ability to develop infrastructure is limited by the limited size of our research group, our limited budget, and the many competing demands for our time. These kinds of limitations are common, so even the techniques that we employ may prove to be useful elsewhere[16].

First, we 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[17]. The result is a pooling of expertise which emphasizes complementary skills and cooperative effort; we help each other. Because of our diverse backgrounds, we 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 probably an answer available[18].

Second, we maintain an extensive library of software documentation: vendors' manuals, third-party tutorials on the major programs, and about 120 pages in six tutorial manuals we have written our 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 our own manuals involve a tradeoff: staff time to write and update them versus learning time saved for new group members (or learning time available to read them). We have found that undergraduate assistants are helpful in making minor updates to these documents, but that the primary writing requires the experience of graduate RAs and the PI.

Third, we 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 our software with less administrative complexity. In particular, there are numerous books for our primary word processor and our relational data base.

Finally, we use informal one-on-one training, conducted by either the PI or one of the senior RAs. At best, individualized tutorials allow us to tailor our training to the specific needs of each individual. At worst, the time we 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. It is another set of tradeoffs, where choices involves guessing about several imprecise factors. How much PI or RA time can be diverted from research and writing? How much will the student contribute to the project (considering background, skills, interests, motivation, and time remaining to complete a degree)? Will the student be able to work independently; what are the risks if supervision is minimal (computers re-configured in surprising ways ...

work that has to be redone ... perhaps important files accidently erased)? How much expertise is really needed to accomplish a task? (Creating a plain text document and restructuring a database requires very different degrees of skill and training time.

Hardware skills

In order to contain costs we try to maintain our 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. We usually evaluate the potential gains of an upgrade against the direct expenses and the likely staff time required to fiddle with equipment.

Even seemingly straightforward modifications can involve unexpected work. One author recently added a second fixed disk drive to his personally owned 8088-based PC. On opening the computer case, he found first that an additional set of cables would be needed (which caused a separate trip to the store). He then found that although space was available for the new drive, the existing mounting bracket was designed only for one drive. The solution was a minor bit of sheet-metal work, but the real cost was time. Overall, installing the disk might have required less than an hour if all had gone well; instead, it took over four hours — and more complex skills.

Our recent upgrade to our 286 based PCs was much larger and more complex. It essentially involved returning our very oldest (XT-style) PCs to the ICS department, replacing in them the disk drives which we had "borrowed" for some of the newer 80286-based clones, and installing a set of even newer 80286 machines which contained larger disk drives. All of the software and data files for each user had to be kept intact throughout the process. We estimated this to be at least a 40-hour job, which our existing staff could not handle. Fortunately, we were able to hire a highly-qualified ICS student who had worked with us for two quarters on a short-term basis. He worked evenings and weekends for four weeks around his full-time job and was unusually skillful in keeping our operation uninterrupted. We have found that the sharpest and most reliable undergraduate students can provide invaluable computer support; but many technical undergraduates are not adequately savvy in computing and not adequately reliable week after week.

Conclusions about Skills

In recruiting RAs, basic research interests and skills are paramount. In contrast we emphasize computing skills in recruiting undergraduate assistants -- both for general office work and computer support. However, all the project team members, including the PI, have significantly enhanced their computing skills in the course of the project. In the case of using key application software, such as word processors and outliners some the skills are co-extensive with the substantive research task. But in the majority of cases, the skills are required to support the task via computers: for example the mechanics of using a statistical package does not teach much about statistical inference. Further we have had to develop a set of procedures for managing a collection of documents, research data and administrative data than students encounter until they participate in a research project. Consequently, we spend significant time teaching all project staff about the pragmatics of file management in a distributed, shared computing environment.

The Complexity and Cost of Computing Support

Many of our routine computer support activities contribute indirectly to our major research goals. But we have found that many of these tasks are essential for containing costs and making computing easy to use. When we ignored them, we incurred even greater hidden costs (e.g., time).

Software Upgrades

No single software package offers enough features to suit every application — nor operates in a way that is convenient for every user. Many organizations have computer aficionados on their staffs who continually search for programs that might make their jobs easier. On this project, the PI and some of the ICS students have continually scanned PC magazines for improved software. Vendors regularly offer new versions of their software, hoping to maintain the loyalty of present customers and to attract new users. These upgrades may simply fix "bugs" that were not found before product release, but in most cases they feature expanded capabilities. An obvious cost that is associated with evolving software is its increase in size. Our file-management utility program occupied 85 KB of storage space in an early version; the present release takes 520 KB. The distribution disk set for a programming language that we use has expanded from one disk to eight during the course of several revisions.

Feature-rich packages are often time-consuming to learn, and more difficult for an occasional user to remember. We try to balance between technical capabilities and ease-of-use (to reduce training costs) both in selecting new applications and in deciding whether or not to install upgraded versions of software that we presently own[19].

The purchase price of a software package is only the beginning of costs associated with its installation and use. First is the theoretically simple task of transferring the program files to hard disk, and making archival back-up copies. Most complex programs today are furnished with "installation" routines that expedite this process. But merely transferring files is not the end of the installation procedure. Other aspects of the computer configuration may have to be modified: for example, default values for the number of files or buffers that the operating system permits. Other programs that are in use (especially those that remain "memory-resident"), may interact with the new program in unpredictable ways — and may have to be modified or deleted. The new program itself may require some "customization": for example, to select the hardware in use and to specify where certain types of data files may be found. Individual users may want to make even more changes, such as in the screen colors used to display information. To save disk storage space, users may want to be selective about the new program files they actually install; some programs are furnished with extensive on-line tutorials that may be initially useful but might be deleted later. We have found that student assistants can be very helpful in performing many of these tasks. But the PI or a senior RA will still spend time verifying the installation, and the entire staff will spend time learning to use it. The total time for these activities has ranged from 1-2 hours for a simple software version update in all of our computers, to 20 hours or more for installing and testing a new software package.

File access and maintenance

The earliest PCs had little or no permanent disk storage; users filed data on individual, small-capacity, diskettes. When 10 MB hard disks first became available, many users began to realize that this much space could not be used for just random file storage. Computer files now had to be organized as carefully as the old pieces of paper, or important documents might become virtually irretrievable. With much larger disk capacity (e.g., 70 MB), and multiple users on a machine, we have found better file organization to be well worth the effort.

We have adopted several complimentary approaches to this task. First, we segregate program and data files into separate directories on the hard disks. In some cases, we have even used separate drives or disk partitions (e.g., "c:" for programs and "d:" for data). Even on machines that have essentially a single user, we establish a directory for that individual's data, with subordinate directories for the various types or subjects of documents, charts, and data files. We have adopted file naming conventions, used by all staff members, to identify for example the word processor with which a document was generated, or the exact version of a data file. On the shared machines with limited disk space, we request each user to limit his or her own data files so that there is a minimum residual "free" space (3MB) on the disk: this prevents unpleasant surprises such as running out of space to execute a program or store the results. We rely heavily on a utility program that enables us to conveniently find, copy, and move files within a complex tree-structured disk directory system.

Most software manuals warn users to periodically "back up" or archive their data files, and we support this wholeheartedly. If we were to lose our survey data, for example, the project simply would be critically harmed. We have found that there are several considerations in creating archives that the manuals seldom address. We found that we reduce confusion and save overall time by organizing the archives (on floppy disks) as carefully as the current data on hard disk. We use a diskette cataloging program as the archive files are created, and find that a conscientious undergraduate assistant can make a large difference in the quality and useability of the archives and the catalog. We use a program to compress many files, saving storage space on the resulting disks[20]. We distribute copies of the archives to multiple locations, both within the office and at the PI's and RAs' homes. Short of renting vault storage, this is our best guard against catastrophic data loss due to fire, earthquake, theft, or vandalism.

Shared access to our administrative data bases and statistical data is a logistical problem which is compounded by our geographical dispersion. We have found it most effective to assign data-base design and maintenance to and statistical data management to specific people using specific computers in the PPRO office. This practice enhances our control over data accuracy but staff lose immediate access to the raw data: paper reports distributed to the staff may not represent the most current information. In addition, we periodically distribute the data files to those staff who need them at their home offices[21].

Multiple formats and multiple users

Unless an organization can operate with a single, totally self-contained computer system, there will be occasions where some kinds of data format conversions are necessary. We have had to convert data both from external sources to our own formats and from one package format to another that we use within our group. Our European collaborator uses a different family of PCs; his data had to be translated for our use not only in written language (French to English), but in computer disk formats. Our own data entry procedure involves the use of a data base package, then a sequence of file exports and imports from the data base to a spreadsheet to the final statistical package format. We produce some printed reports by the opposite process: statistical package to spreadsheet to data base program for the final sorting and output. To use ICS Sequent mainframe, we must either translate files from our word processor format to pure "ASCII text," or we must use special binary file communication commands to preserve the format of word processor documents as they are transmitted. These procedures represent more hidden costs in both staff efforts to solve conversion problems and in time to perform the actual conversions.

Within the office, our multiple-user computing environment requires tradeoffs to be made between standardized, easy-for-everyone-to-use procedures and flexibility to meet individuals' computing needs and preferences. We have already described the "core" of software — major programs and utilities — that we have installed on all machines in the office. We also tailor certain machines to specific applications: for example, one computer contains "soft fonts" and other utilities required to operate the laser printer. We have recently adopted an "inventory" policy, in which we check each machine at the beginning and end of each academic term — insuring that all software is up to date, utilities (e.g., disk optimization) are run, and unneeded files are expunged.

Hardware Maintenance

Sometimes a hardware upgrade is unavoidable. Our statistical software simply will not run efficiently without a math co-processor, which does not come as standard equipment with most PCs. Programs and data files require storage space; as they expand, so must the available disk capacity. Other "improvements" may be more difficult to evaluate: a faster processor might still spend the majority of time waiting for user input. 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. We usually evaluate the potential gains of an upgrade against the direct expenses and the likely staff time required to fiddle with equipment.

Computer upgrades can be planned in advance while repairs, are usually required because equipment fails unexpectedly. We have experienced disk drive failures, erratic printer operation, random failure of computers to start up or "boot" properly, and a tendency of some PCs to make unusual and irritating "squeaking" noises. Each of these problems has cost RA or undergraduate time for troubleshooting. In the last case we spent time purchasing and installing a new power supply —- a plausible solution, but ineffective in this case.

Our most effective alternative has normally been to contract with outside repair shops for serious hardware repairs, on a case-by-case basis. We perceived long-term maintenance contracts or "extended warranties" as too costly for the amount of services that we actually needed. We have occasionally used the services of the campus computing facility maintenance staff on a charge-back basis. We also spend some time on regular preventive maintenance. This work may be as simple as periodically running a disk-maintenance program or a program designed to detect any "virus infection" that might have been caused by new programs.

Computer system administration

Maintaining the configuration of all of our PCs is not a trivial task. We rely on an ICS undergraduate "system administrator" to provide support for our computing milieu in exchange for special study course credit. Designing this job is a delicate balance so that it has legitimate academic value to the students and also helps leverage our time. This student normally devotes about ten hours per week to these activities, and works under the supervision of a graduate RA with from ICS or the PI. Most system administrators have worked for us for two quarters, since it takes them much of one quarter to effectively learn our environment. Occasionally we 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 tries to channel 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.

Conclusions about support

Support is the hidden soft underbelly of computerization. We have estimated the overall time required for computer support and maintenance within our project 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 we believe that these figures are very conservative[22]. 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 our time budget – a significant commitment.

CONCLUSIONS

This paper has examined the variety of skills and time required to support research computing in a PC-oriented research group. While many researchers will recognize from their own activities some of the experiences that we have described, they may rarely evaluate these experiences in a systematic way. We believe that the insights that we have developed here will be useful to other research groups.

We have examined the dynamics of computerization in a specific research group. This analysis pays special attention to the organization of academic work and resource politics on the campus, and the commitments of research staff to multiple organizations and multiple lines of work, as well as the information processing capabilities of computing technologies. This analysis illustrates the ways that computerization projects are integrated into a set of labor processes, even though they may alter some of their aspects.

We have illustrated how the commitments to particular computing configurations made early in the research project constrained the teams' subsequent developmental trajectory. Specific choices of equipment and staffing were situated responses to carrying out key research tasks while bargaining for computing resources in a social setting with a specific configuration of interests, computing equipment, and human resources. The routine link to other groups, in this case ICS and PPRO, also influenced the political economy of choice. Our approaches were not simply optimal strategies for matching equipment to tasks that would make sense regardless of the structuring of resources and interests in the larger social setting.

However, we have also shown that there is a much wider range of costs associated with computerization than vendors, the popular literature and the professional literatures address. These findings give depth to King and Kraemer's (1981) observation that decentralized computing has many costs that are not apparent to central administrators. 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 that the PI and staff members devote to training new staff, and upgrading and maintaining computing, and they are frequently difficult to quantify precisely. With limited budget, a research 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.

We have emphasized the central role of computing infrastructure. In a small research group, it is likely that this infrastructure will be informal. An opportunistic team can scavenge their environment for computing support. In he case of this group, the team relied upon undergraduate students to help support computing. This was a delicate strategy that had few direct costs, but which required significant faculty time to make it workable and to insure that students educational preferences were satisfied, as well as meeting the operational demands of the project.

Finally, we have suggested that computing infrastructure may be enhanced by careful organization. Specific techniques that we have found to be useful include assigning single points of responsibility for key tasks such as data base management and software configuration; adopting uniform disk file structuring, naming, and archiving conventions; and providing a standardized core of software which group members may refine to meet their own research requirements and work preferences. We have learned to be cautious about refinements: we find that the seduction of new software and hardware, and the flexibility to continuously "improve" our output, sometimes leads us to spend much more time on refinements than would be optimal in retrospect.

New technologies will give us more computing power for modest costs in the future. But despite improvements in the information processing capabilities of new technologies, social processes will continue to play a significant role in the ways that people work with and around computing.

ACKNOWLEDGEMENTS

This research was funded by National Science Foundation grant #IRI-87-09613. We have benefitted from continuing discussions about computing issues and the role of computing support with our colleagues and associates: Jonathan Allen, Joey George, Suzanne Iacono, Kate Kaiser, John King, Kenneth Kraemer, Steve Lepore. A very special thanks to Jim Brennan, Alfredo Con, Ken Lloyd, Bill Pike, Mark Raymond, and

Hanny Suryadharma — who kept our computers running! Mary Zmuidzinas and the anonymous referees provided very helpful comments which sharpened our final manuscipt.

NOTES

Tom Jewett is a research associate at UC-Irvine's Public Policy Research Organization, and an adjunct faculty member at Chapman College and National University. Rob Kling is Professor of Information and Computer Science and Management at UC-Irvine. He has published numerous articles about which examines the social aspects of computing and is co-editor of *Computerization and Controversy: Value Conflicts and Social Choices*.

1. Rob Kling and Suzanne Iacono developed the Desktop Computing Research Project which is the subject of this study in 1985. Tom Jewett joined the project in January 1989 and worked regularly with the project equipment through July 1990. In addition, he supervised many of the undergraduate students whose work is discussed. Data from this paper comes from notes and memos by the project staff as well as recollections by the authors. In addition the authors drew on two empirical studies of the project done by people who were not members of the team. Debra Morrison and Marielle Steinke interviewed all the project members in 1988 and wrote a paper about the development of computing on the project and key problem areas. In the Spring of 1989, Linda Kao conducted a similar study.

2. Much more detailed accounts of the Desktop Computing Research Project are given in (Jewett and Kling, 1990; Lepore, Kling, Iacono and George, 1989; and George, Kling and Iacono, 1989). For further information please contact Prof. Rob Kling (email kling@ics.uci.edu).

3. We carefully distinguish between the paid undergraduates and the academic-credit students when we assign duties to them. Those who are paid might do repetitive tasks such as data entry; the others will do projects with specific educational goals (for example, learning to manage a complex data environment).

4. One indicator of our skill comes from our survey data — which we administered to ourselves. We were among the highest-ranking work groups in in-house computing expertise. We also ranked highly in access to computing resources, worker participation in decisions about computing and work life, and flexibility of working conditions. We selected work groups that were highly automated to participate in our study. Their primary computing activities varied; some emphasized word processing, some record-keeping, and some data analysis. Our own computing environment is similar to the data analysis groups in complexity.

5. The individual-level file is a composite of 20-30 separate data-entry files. Identifying, consolidating, and archiving these files is in itself a data-management issue for our analysis staff.

6. IBM, IBM-PC, PC-XT, and PC-AT are registered trademarks of the International Business Machines Corporation. We will use the abbreviation "PC" (by itself) in the colloquial sense to mean any "personal" (micro) computing equipment.

7. We will discuss the choices leading to this change in a later section.

8. Some universities, for example, may provide laser-quality printing of computerized documents on a "walk-in" basis that is similar to copy machine access elsewhere. We are not fortunate enough to have this particular service available.

9. Even this apparent good fortune caused dilemmas for us. In our initial tests, the laser printers were as slow as dot matrix printers. The problem turned out to be an incompatibility between our memory-resident print spooler program and the output program for the laser printer.

10. Anderson and Brent (1989) seem to view as progress the increasing receptiveness of journals to electronic submission of articles, and predict that this will become a requirement in the near future. We view the procedure more skeptically: it may save time -- but many of the burdens of production are now distributed from the editors to individual research teams, who may or may not be equipped to handle them efficiently.

11. The PI also used this database program in his teaching. Consequently some of the undergraduates we recruited for the project were familiar with it.

12. We attribute these problems both to the complexity of the data domain and to our informal methods of handling the development. Lacking resources for formal data-base analysis, the PI encourages staff members to "make the system work" as well as possible within their own skill level and understanding of the task.

13. We have encountered few minor problems in our email procedures, but the medium has a special quality. There is the lack of feedback and "body language" of a face-to-face discussion — or even the voice inflection of a telephone call. We cope with these limitations, as others do, liberally sprinkling our messages with explicit cues such as (grin), (chuckle), or (mount soapbox), and sometimes inventing unique spelling and punctuation rules to emphasize a point.

14. Undergraduate students are not used to working in a setting where they share data and computers with other people whose work schedules they have to work around and system configurations they can't casually customize without negotiation.

15. Our survey results show us to be only mediocre in this area compared to other work groups in the study.

16. We also realize that there are organizations with even more rudimentary procedures. When one of the authors joined the computer science department on campus, he was told: "Your mainframe account is set up. Log on with your last name; the password is 'changeme.' If you have any questions, just type 'man -k.'" One of our colleagues reported the same experience; that was the extent of the training program! (Note: "man" is the command on this computer which accesses the on-line manual. Typed by itself, it produces the cryptic message "Usage: man [section] name..."; typed with the "-k" option, it produces the even more cryptic "apropos what?".

17. This is confirmed in our flexibility and participation scores from the survey. We appear to fit the

management paradigm that suggests that research groups (for example) require a more flexible approach than organizations whose tasks are repetitive (such as data entry).

18. In this respect, we are similar to many other work groups in our study, who frequently find the informal computing infrastructure to be more effective than the formal one (George, Kling, and Iacono, 1989; Jewett and Kling, 1990).

19. Product reviews in journals such as this one and in the popular computer-oriented press are valuable in assessing new products. Colleagues' experiences with specific packages can be extremely useful as a guide. Vendors occasionally offer "trial" versions of their software that enable an organization to evaluate it under their own conditions without committing to the full expense of a purchase. We have employed all of these methods in selecting our current software suite -- and we will continue to evaluate new products as they are released.

20. For example, our statistical data files are often compressed by 80% (from around 600 KB to 120 KB).

21. Some of our students proposed that we connect all of our PCs in a local-area network, and maintain the data base on a central file server. We had a team of students investigate this option: it would neither help those working at home nor fit our budget.

22. This estimate excludes time spent by project staff members in initial training or in learning new software packages or new features of software needed to do their research tasks (e.g., statistical analyses, specialized word processor formatting).

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

Kraemer, Kenneth L., John L. King, Debora E. Dunkle, and Joseph P. Lane. (1989). *Managing Information Systems: Change and Control in Organizational Computing*. San Francisco. Jossey-Bass.

Laudon, Kenneth. (1986). "From PCs to Managerial Workstations: Organizational Environment and Management Policy in the Financial Industry." in Matthias Jarke (Ed.) *Managers, Micros and Mainframes: Integrating Systems for End-Users*. New York, John Wiley.

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

Poppel, Harvey E. (1982). Who Needs the Office of the Future? *Harvard Business Review,* (Nov-Dec):146-155.

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

Treibwasser, Marc A. (1987). Social Science and the New Generation of Microcomputer Hardware. *Social Science Microcomputer Review,* 5:3, Fall, pp 289-301.

Walton, Richard (1989). *Up and Running: Integrating Information Technology and the Organization.* Boston, Ma: Harvard University Business School Press.

Ward, John, Pat Griffiths, and Paul Whitmore. (1990). *Strategic Planning for Information Systems*. New York, John Wiley.

REFERENCES

Anderson, Ronald E., and Edward E. Brent (1989). Computing in Sociology: Promise and Practice. *Social Science Computer Review,* 7:4, Winter, pp. 487-502.

Bullen, Christine and John Bennett. (in press) "Groupware in Practice: An Interpretation of Work Experience" in Dunlop and Kling (in press).

Clement, Andrew. (1990). Cooperative Support for Computer Work: A Social Perspective on Empowering End Users. *Proc. 1990 Conference on Computer Supported Cooperative Work*: 223-236.

Dunlop, Charles and Rob Kling. (in press). "Computerization and The Transformation of Work" in *Computerization and Controversy: Value Conflicts and Social Choices* Charles Dunlop and Rob Kling (Ed.) San Diego. Academic press.

Gasser, Les. (1986). "The Integration of Computing and Routine Work." *ACM Transactions on Office Information Systems*, Vol. 4, No. 3 (July):205-225.

George, Joey, Rob Kling, and Suzanne Iacono (1989). The Role of Training and Support in Desktop Computing. *Proceedings of IFIP Workshop on Desktop Computing,* Ithica, NY; Kate Kaiser (ed.). Amsterdam: North-Holland.

Giuliano, Vincent (1982). The Mechanization of Office Work. *Scientific American,* Vol. 247, pp. 148-164.

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* (ACM-SIGOIS Bulletin Vol 11 #2,3), Cambridge, MA, April 25-27, pp. 69-79.

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.

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

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

Kling, Rob (1990). Reading "All About" Computerization: Five Common Genres of Social Analysis. *Directions in Advanced Computing, 1990*; Doug Schuler (ed.). Norwood, NJ: Ablex.

Kling, Rob, and Walt Scacchi (1982). The Web of Computing: Computer Technology as Social Organization. *Advances in Computers,* 21:1-90.

Kling, Rob and Suzanne Iacono. (1984). "The Control of Information Systems Development After Implementation" *Communications of the ACM, 27*(12)(December):1218-1225.

Kling, Rob and Suzanne Iacono. (1988)."The Mobilization of Support for Computerization: The Role of Computerization Movements" *Social Problems*, 35(3)(June):226-243.

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