

Organizational Analysis in Computer Science

Rob Kling

School of Library and Information Science

10th and Jordan, Library 012

Indiana University

Bloomington, IN 47405

Tel: (812)857-7255

Kling@indiana.edu

Revised April 1994 (v. 13.3)

Appears in The Information Society, 9(2) (Mar-Jun, 1993):71-87.

ABSTRACT

Computer Science is hard pressed in the US to show broad utility to help justify billion dollar research programs and the value of educating well over 40,000 Bachelor of Science and Master of Science specialists annually in the U.S. The Computer Science and Telecommunications Board of the U.S. National Research Council has recently issued a report, "Computing the Future (Hartmanis and Lin, 1992)" which sets a new agenda for Computer Science. The report recommends that Computer Scientists broaden their conceptions of the discipline to include computing applications and domains to help understand them. This short paper argues that many Computer Science graduates need some skills in analyzing human organizations to help develop appropriate systems requirements since they are trying to develop high performance computing applications that effectively support higher performance human organizations. It is time for academic Computer Science to embrace organizational analysis (the field of Organizational Informatics) as a key area of research and instruction.

INTRODUCTION

Computer Science is being pressed on two sides to show broad utility for substantial research and educational support. For example, the High Performance Computing Act will provide almost two billion dollars for research and advanced development. Its advocates justified it with arguments that specific technologies, such as parallel computing and wideband nets, are necessary for social and economic development. In the US, Computer Science academic programs award well over 30,000 Bachelor of Science (BS) and almost 10,000 Master of Science (MS) degrees annually. Some of these students enter PhD programs and many work on projects which emphasize mathematical Computer Science. But many of these graduates also take computing jobs for which they are inadequately educated, such as helping to develop high performance computing applications to improve the performance of human organizations.

These dual pressures challenge leading Computer Scientists to broaden their conceptions of the discipline to include an understanding of key application domains, including computational science and commercial information systems. An important report that develops this line of analysis, "*Computing the Future*" (CTF) (Hartmanis and Lin, 1992), was recently issued by the Computer Science and Telecommunications Board of the U.S. National Research Council.

CTF is a welcome report that argues that academic Computer Scientists must acknowledge the driving forces behind the substantial Federal research support for the discipline. The explosive growth of computing and demand for CS in the last decade has been driven by a diverse array of applications and new modes of computing in diverse social settings. *CTF* takes a strong and useful position in encouraging all Computer Scientists to broaden our conceptions of the discipline and to examine computing in the context of interesting applications.

CTF's authors encourage Computer Scientists to envision new technologies in the social contexts in which they will be used. They identify numerous examples of computer applications in earth science, computational biology, medical care, electronic libraries and commercial computing that can provide significant value to people and their organizations. These assessments rest on concise and tacit analyses of the likely design, implementation within organizations, and uses of these technologies. For example, *CTF*'s stories of improved computational support for modelling are based on rational models of organizational behavior. They assume that professionals, scientists, and policy-makers use models to help improve their decisions. But what if organizations behave differently when they use models? For example suppose policy makers use models to help rationalize and legitimize decisions which are made without actual reference to the models?

One cannot discriminate between these divergent roles of modelling in human organizations based upon the intentions of researchers and system designers. The report tacitly requires that the CS community develop reliable knowledge, based on systematic research, to support effective analysis of the likely designs and uses of computerized systems. *CTF* tacitly requires an ability to teach such skills to CS practitioners and students. Without a disciplined skill in analyzing human organizations, Computer Scientists' claims about the usability and social value of specific technologies is mere opinion, and bears a significant risk of being misleading. Further, Computer Scientists who do not have refined social analytical skills sometimes conceive and promote technologies that are far less useful or more costly than they claim. Effective CS practitioners who "compute for the future" in organizations need some refined skills in organizational analysis to understand appropriate systems requirements and the conditions that transform high performance computing into high performance human organizations. Since *CTF* does not spell out these tacit implications, I'd like to explain them here.

BROADENING COMPUTER SCIENCE: FROM COMPUTABILITY TO USABILITY

The usability of systems and software is a key theme in the history of CS. We must develop theoretical foundations for the discipline that give the deepest insights in to what makes systems usable for various people, groups and organizations. Traditional computer scientists commonly refer to mathematics as the theoretical foundations of CS. However, mathematical formulations give us limited insights into understanding why and when some computer systems are more usable than others.

Certain applications, such as supercomputing and computational science are evolutionary extensions of traditional scientific computation, despite their new direction with rich graphical front ends for visualizing enormous mounds of data. But other, newer modes of computing, such as networking and microcomputing, change the distribution of applications. While they support traditional numerical computation, albeit in newer formats such as spreadsheets, they have also expanded the diversity of non-numerical computations. They make digitally represented text and graphics accessible to tens of millions of people.

These technological advances are not inconsistent with mathematical foundations in CS, such as Turing machine formulations. But the value of these formats for computation is not well conceptualized by the foundational mathematical models of computation. For example, text editing could be conceptualized as a mathematical function that transforms an initial text and a vector of incremental alterations into a revised text. Text formatting can be conceptualized as a complex function mapping text strings into spatial arrays. These kinds of formulations don't help us grasp why many people find "what you see is what you get" editors as much more intuitively appealing than a system that links line editors, command-driven formatting languages, and text compilers in series.

Nor do our foundational mathematical models provide useful ways of conceptualizing some key advances in even more traditional elements of computer systems such as operating systems and database systems. For example, certain mathematical models underlie the major families of database systems. But one can't rely on mathematics alone to assess how well networks, relations, or object-entities serve as representations for the data stored in an airline reservation system. While mathematical analysis can help optimize the efficiency of disk space in storing the data, they can't do much to help airlines understand the kinds of services that will make such systems most useful for reservationists, travel agents and even individual travellers. An airline reservation system in use is not simply a closed technical system. It is an open socio-technical system (Hewitt, 1986; Kling, 1992). Mathematical analysis can play a central role in some areas of CS, and an important role in many areas. But we cannot understand important aspects of usability if we limit ourselves to mathematical theories.

The growing emphasis of usability is one of the most dominant of the diverse trends in computing. The usability tradition has deep roots in CS, and has influenced the design of programming languages and operating systems for over 25 years. Specific topics in each of these areas also rest on mathematical analysis which Computer Scientists could point to as "the foundations" of the respective subdisciplines. But Computer Scientists envision many key advances as design conceptions rather than as mathematical theories. For example,

integrated programming environments ease software development. But their conception and popularity is not based on deeper formal foundations for programming languages. However, the growth of non-numerical applications for diverse professionals, including text processing, electronic mail, graphics, and multimedia should place a premium on making computer systems relatively simple to use. Human Computer Interaction (HCI) is now considered a core subdiscipline of CS.

The integration of HCI into the core of CS requires us to expand our conception of the theoretical foundations of the discipline. While every computational interface is reducible to a Turing computation, the foundational mathematical models of CS do not (and could not) provide a sound theoretical basis for understanding why some interfaces are more effective for some groups of people than others. The theoretical foundations of effective computer interfaces must rest on sound theories of human behavior and their empirical manifestations (cf. Ehn, 1991, Grudin, 1989).

Interfaces also involve capabilities beyond the primary information processing features of a technology. They entail ways in which people learn about systems and ways to manage the diverse data sets that routinely arise in using many computerized systems (Kling, 1992). Understanding the diversity and character of these interfaces, that are required to make many systems usable, rests in an understanding the way that people and groups organize their work and expertise with computing. Appropriate theories of the diverse interfaces that render many computer systems truly useful must rest, in part, on theories of work and organization. There is a growing realization, as networks tie users together at a rapidly rising rate, that usability cannot generally be determined without our considering how computer systems are shaped by and also alter interdependencies in groups and organizations. The newly-formed subdiscipline of Computer Supported Cooperative Work and newly-coined terms "groupware" and "coordination theory" are responses to this realization (Greif, 1988; Galegher, Kraut and Egido, 1990).

BROADENING COMPUTER SCIENCE: FROM HIGH PERFORMANCE COMPUTING TO HIGH PERFORMANCE ORGANIZATIONS

The arguments of *CTF* go beyond a focus on usable interface designs to claims that computerized systems will improve the performance of organizations. The report argues that the US should invest close to a billion dollars a year in CS research because of the resulting economic and social gains. These are important claims, to which critics can seek systematic evidence. For example, one can investigate the claim that 20 years of major computing R&D and corporate investment in the US has helped provide proportionate economic and social value.

CTF is filled with numerous examples where computer-based systems provided value to people and organizations. The tough question is whether the overall productive value of these investments is worth the overall acquisition and operation costs. While it is conventional wisdom that computerization must improve productivity, a few researchers began to see systemic possibilities of counter-productive computerization in the early 1980s (King and Kraemer, 1981). In the last few years economists have found it hard to give

unambiguously affirmative answers to this question. The issue has been termed "The Productivity Paradox," based on a comment attributed to Nobel laureate Robert Solow who remarked that "computers are showing up everywhere except in the [productivity] statistics (Dunlop and Kling, 1991a)."

Economists are still studying the conditions under which computerization contributes to organizational productivity, and how to measure it [1]. But even if computerization proves to be a productive investment, in the net, in most economic sectors, there is good reason to believe that many organizations get much less value from their computing investments than they could and should.

There is no automatic link between computerization and improved productivity. While many computer systems have been usable and useful, productivity gains require that their value exceed all of their costs.

There are numerous potential slips in translating high performance computing into cost-effective improvements in organizational performance. Some technologies are superb for well-trained experts, but are difficult for less experienced people or "casual users." Many technologies, such as networks and mail systems, often require extensive technical support, thus adding hidden costs (Kling, 1992).

Further, a significant body of empirical research shows that the social processes by which computer systems are introduced and organized makes a substantial difference in their value to people, groups and organizations (Lucas, 1981; Kraemer, et. al. 1985; Orlikowski, 1992). Most seriously, not all presumably appropriate computer applications fit a person or group's work practices. While they may make sense in a simplified world, they can actually complicate or misdirect real work.

Group calendars are but one example of systems that can sound useful, but are often useless because they impose burdensome record keeping demands (Grudin, 1989). In contrast, electronic mail is one of the most popular applications in office support systems, even when other capabilities, like group calendars, are ignored (Bullen and Bennett, 1991). However, senders are most likely to share information with others when the system helps provide social feedback about the value of their efforts or they have special incentives (Sproull and Kiesler, 1991; Orlikowski, 1992). Careful attention to the social arrangements or work can help Computer Scientists improve some systems designs, or also appreciate which applications may not be effective unless work arrangements are changed when the system is introduced.

The uses and social value of most computerized systems can not be effectively ascertained from precise statements of their basic design principles and social purposes. They must be analyzed within the social contexts in which they will be used. Effective social analyses go beyond accounting for formal tasks and purposes to include informal social behavior, available resources, and the interdependencies between key groups (Cotterman and Senn, 1992).

Many of the BS and MS graduates of CS departments find employment on projects where improved computing should enhance the performance of specific organizations or industries.

Unfortunately, few of these CS graduates have developed an adequate conceptual basis for understanding when information systems will actually improve organizational performance. Consequently, many of them are prone to recommend systems-based solutions whose structure or implementation within organizations would be problematic.

ORGANIZATIONAL INFORMATICS

Organizational Informatics denotes a field which studies the development and use of computerized information systems and communication systems in organizations. It includes studies of their conception, design, effective implementation within organizations, maintenance, use, organizational value, conditions that foster risks of failures, and their effects for people and an organization's clients. It is an intellectually rich and practical research area.

Organizational Informatics is a relatively new label. In Europe, the term Informatics is the name of many academic departments which combine both CS and Information Systems. In North America, Business Schools are the primary institutional home of Information Systems research and teaching. But this location is a mixed blessing. It brings IS research closer to organizational studies. But the institutional imperatives of business schools lead IS researchers to emphasize the development and use of systems in a narrow range of organizations -- businesses generally, and often service industry firms. It excludes information systems in important social sectors such as health care, military operations, air-traffic control, libraries, home uses, and so on. And Information Systems research tries to avoid messy issues which many practicing Computer Scientists encounter: developing requirements for effective systems and mitigating the major risks to people and organizations who depend upon them.

The emerging field of Organizational Informatics builds upon research conducted under rubrics like Information Systems and Information Engineering. But it is more wide ranging than either of these fields are in practice[2].

Organizational Informatics Research

In the last 20 years a loosely organized community of some dozens of researchers have produced a notable body of systematic scientific research in Organizational Informatics. These studies examine a variety of topics, including:

- * how system designers translate people's preferences into requirements;
- * the functioning of software development teams in practice;
- * the conditions that foster and impede the implementation of computerized systems within organizations;
- * the ways that computerized systems simplify or complicate coordination within and between organizations;
- * how people and organizations use systems in practice;

* the roles of computerized systems in altering work, group communication, power relationships, and organizational practices.

Researchers have extensively studied some of these topics, such as computerization and changing work, appear in synoptic review articles (Kling and Dunlop, in press). In contrast, researchers have recently begun to examine other topics, such software design (Winograd and Flores, 1986; Kyng and Greenbaum, 1991), and have recently begun to use careful empirical methods (e.g. Suchman, 1983; Bentley, et. al, 1992; Fish, et. al., 1993). I cannot summarize the key theories and rich findings of these diverse topics in a few paragraphs. But I would like to comment upon a few key aspects of this body of research.

Computer Systems Use in Social Worlds

Many studies contrast actual patterns of systems design, implementation, use or impacts with predictions made by Computer Scientists and professional commentators. A remarkable fraction of these accounts are infused with a hyper-rational and under-socialized view of people, computer systems, organizations and social life in general. Computer Scientists found that rule driven conceptions to be powerful ways to abstract domains like compilers. But many Computer Scientists extend them to be a tacit organizing frame for understanding whole computer systems, their developers, their users and others who live and work with them. Organizations are portrayed as generally cooperative systems with relatively simple and clear goals. Computer systems are portrayed as generally coherent and adequate for the tasks for which people use them. People are portrayed as generally obedient and cooperative participants in a highly structured system with numerous tacit rules to be obeyed, such as doing their jobs as they are formally described. Using data that is contained in computer systems, and treating it as information or knowledge, is a key element of these accounts. Further, computer systems are portrayed as powerful, and often central, agents of organizational change.

This Systems Rationalist perspective infuses many accounts of computer systems design, development, and use in diverse application domains, including CASE tools, instructional computing, models in support of public policy assessments, expert systems, groupware, supercomputing, and network communications (Kling, 1980; Kling, Scherson and Allen, 1992).

All conceptual perspectives are limited and distort "reality." When Organizational Informatics researchers systematically examine the design practices in particular organizations, how specific groups develop computer systems, or how various people and groups use computerized systems, they find an enormous range of fascinating and important human behavior which lies outside the predictive frame of Systems Rationalism (Kling, 1980). Sometimes these behaviors are relatively minor in overall importance. But in many cases they are so significant as to lead Organizational Informatics researchers to radically reconceptualize the processes which shape and are shaped by computerization.

There are several alternative frames for reconceptualizing computerization as alternatives to Systems Rationalism. The alternatives reflect, in part, the paradigmatic diversity of the social sciences. But all of these reconceptions situate computer systems and organizations in richer social contexts and with more complex and multivalent social relations than does

systems rationalism. Two different kinds of observations help anchor these abstractions.

Those who wish to understand the dynamics of model usage in public agencies must appreciate the institutional relationships which influence the organization's behavior. For example, to understand economic forecasting by the US Congress and the U.S. Executive branch's Office of Management and Budget, one must appreciate the institutional relations between them. They are not well described by Systems Rationalist conceptions because they were *designed* to continually differ with each other in their perspectives and preferred policies. That is one meaning of "checks and balances" in the fundamental design of the US Federal Government. My colleagues, Ken Kraemer and John King, titled their book about Federal economic modelling, *DataWars* (Kraemer, et. al., 1985). Even this title doesn't make much sense within a Systems Rationalist framework.

Modelling can be a form of intellectual exploration. It can also be a medium of communication, negotiation, and persuasion. The social relationships between modelers, people who use them and diverse actors in Federal policymaking made these socially mediated roles of models sometimes most important. In these situations, an alternative view of organizations as coalitions of interest groups was a more appropriate conceptualization. And within this coalitional view of organizations, a conception of econometric models as persuasion support systems rather than as decision support systems sometimes is most appropriate. Organizational Informatics researchers found that political views of organizations and systems developments within them apply to many private organizations as well as to explicitly political public agencies.

Another major idea to emerge from the broad body of Organizational Informatics research is that the social patterns which characterize the design, development, uses and consequences of computerized systems are dependent on the particular ecology of social relationships between participants. This idea may be summarized by saying that the processes and consequences of computerization are "context dependent." In practice, this means that the analyst must be careful in generalizing from one organizational setting to another. While data wars might characterize econometric modelling on Capitol Hill, we do not conclude that all computer modelling should be interpreted as persuasion support systems. In some settings, models are used to explore the effects of policy alternatives without immediate regard for their support as media for communication, negotiation or persuasion. At other times, the same model might be used (or abused with cooked data) as a medium of persuasion. The brief accounts of models for global warming in *CTF* fit a Systems Rationalist account. Their uses might appear much less "scientific" if they were studied within the actual policy processes within which they are typically used.

Computing in a Web of Technological and Social Dependencies: The Role of Infrastructure

Another key feature of computerized systems is the technological and organizational infrastructure required to support their effective use (Kling and Scacchi, 1982; Kling, 1987; Kling, 1992). The information processing models of computerized systems focus on the "surface structures," such as information flows within a system. For example, one can compare the information processing capabilities of computerized modelling systems in

terms of the complexity and variety of computations that they support, the richness of their graphical displays, and so on. Text processing systems can be similarly compared by contrasting their capabilities for handling footnotes, graphics, fine grained text placement, custom dictionaries and so on. From an information processing point of view, system A is usually better than system B if it offers many more capabilities than system B. Information processing conceptions have also fueled much of the talk about high performance computing. It is common to talk about massively parallel computing in terms of the scale and unit cost of computation (Kling, Scherson, and Allen 1992), and the discussions of networking in terms of the wide data bandwidths that new technologies offer.

If we ask how these technologies improve organizational performance, then we have to ask how they can be made usable to diverse groups. The most powerful modelling system may be of limited utility if it requires sophisticated programming skills to create and modify every data transformation. Alternatively, such a package can be made more widely useful by having the modelling efforts managed by a programming group which provides added value for added cost.

Few people are capable or interested in primarily using "raw computing" for their work. The diverse array of "productivity software" -- such as text processing, presentation graphics, spreadsheets, databases and so on gain their value when they can be provided and maintained in a way that matches the skills and available time of people who will use them. Both skill and time are scarce resources in most organizations. Skilled time is especially expensive.

Similarly, the organizational value of digital libraries can't be adequately conceptualized in terms of simple data-centric measures, like the number of gigabytes of available files. The ease of people accessing useful documents is much more pertinent, although much less frequently discussed today.

In each of these cases, the support systems for the focal computing system is integral to the effective operation of the technology. Infrastructure refers to the set of human and organizational resources that help make it simpler and faster for skilled people to use computerized systems. Infrastructure should be part of the conceptualization. Often the support systems for a computing can involve several different organizations, including hardware and software vendors, telecommunication support groups, divisional systems groups, and local experts (Kling, 1992). It can be organizationally very complex and unresponsive in some cases and organizationally simpler and more effective in others. In any case, the infrastructure for systems support can't be ignored when one is interested in improving organizational performance.

Repercussions for Systems Design

Even when computerized systems are used as media of intellectual exploration, Organizational Informatics researchers find that social relationships influence the ways that people use computerized systems. Christine Bullen and John Bennett (1991) studied 25 organizations that used groupware with diverse modules such as databases, group calendars, text annotating facilities and electronic mail. They found that the electronic mail modules were almost universally valued, while other system facilities were often unused.

In a recent study, Sharyn Ladner and Hope Tillman examined the use of the Internet by university and corporate librarians. While many of them found data access through databases and file transfer to be important services, they also reported that electronic mail was perhaps the most critical Internet feature for them.

The participants in our study tell us something that we may have forgotten in our infatuation with the new forms of information made available through the Internet. And that is their need for community. To be sure, our respondents use the Internet to obtain information not available in any other format, to access databases ... that provide new efficiencies in their work, new ways of working. But their primary use is for communication. Special librarians tend to be isolated in the workplace -- the only one in their subject specialty (in the case of academe), or the only librarian in their organization (in the case of a corporate library). Time and time again our respondents expressed this need to talk to someone -- to learn what is going on in their profession, to bounce ideas off others, to obtain information from people, not machines. There are tremendous implications from the Internet technology in community formation -- the Internet may indeed provide a way to increase community among scholars, including librarians. The danger we face at this juncture in time, as we attach library resources to the Internet, is to focus all of our energies on the machine-based resources at the expense of our human-based resources, i.e., ourselves (Ladner and Tillman, 1992).

In these studies, Organizational Informatics researchers have developed a socially rich view of work with and around computing, of computing within a social world. These studies have strong repercussions for the design of software. A good designer cannot assume that the majority of effort should go into the "computational centerpiece" of a system, while devoting minor efforts to supporting communication facilities. One of my colleagues designed a modelling system for managers in a major telephone company, after completing an extensive requirements analysis. However, as an afterthought, he added a simple mail system in a few days work. He was surprised to find that the people who used these systems regularly used his crude electronic mail system, while they often ignored interesting modelling capabilities. Such balances of attention also have significant repercussions. Many people need good mail systems, not just crude ones: systems which include facile editors, ease in exporting and importing files, and effective mail management (Kling and Covi, 1993).

Assessing people's preferences for systems' designs is an exercise in social inquiry. While rapid prototyping may help improve designs for some systems, it is less readily applicable to systems which are used by diverse groups at numerous locations. Computer scientists are beginning to develop more reliable methods of social inquiry to better understand which systems designs will be most useful (Bentley, et. al. 1992; Kyng and Greenbaum, 1991). It

is particularly helpful to organize system designs that help minimize the complexity and cost of its infrastructure (Kling, 1992).

Fish and his colleagues (1993) recently reported the way that the explicit use of social theory helped them design more effective group meeting systems. Unfortunately, these newer methods are rarely taught to CS students. When computer specialists build an imbalanced system, it should not be a surprise when the resulting organizational value of their efforts is very suboptimal.

System Security and Reliability

In a simplified engineering model of computing, the reliability of products is assured through extensive testing in a development lab. The social world of technology use not perceived as shaping the reliability of systems, except through irascible human factors, such as "operator errors." An interesting and tragic illustration of the limitations of this view can be found in some recent studies of the causes of death and maiming by an electron accelerator which was designed to help cure cancer, the Therac-25 (Jacky, 1991, Leveson and Turner, 1993).

The Therac-25 was designed and marketed in the mid 1980s by a Canadian firm, Atomic Energy of Canada Limited (AECL), as an advanced medical technology. It featured complete software control over all major functions (supported by a DEC PDP-11), among other innovations. Previous machines included electro-mechanical interlocks to raise and lower radiation shields. Several thousand people were effectively treated with the Therac-25 each year. However, between 1985 and 1987 there were six known accidents in which several people died in the US. Other were seriously maimed or injured [3].

Both studies concur that there were subtle but important flaws in the design of the Therac-25's software and hardware. AECL's engineers tried to patch the existing hardware and (finally) software when they learned of some of the mishaps. But they treated each fix as the final repair.

Both studies show how the continuing series of mishaps was exacerbated by diverse organizational arrangements. Jacky claims that pressures for speedy work by radiological technicians coupled with an interface design that did not enhance important error messages was one of many causes of the accidents. Leveson and Turner differ in downplaying the working conditions of the Therac-25's operators and emphasize the flawed social system for communicating the seriousness of problems to Federal regulators and other hospitals. Both studies observe that it is unlikely for the best of companies to develop perfect error-free systems without high quality feedback from users. Their recommendations differ: Jacky discusses the licensing of system developers and the regulation of computerized medical systems to improve minimal standards of safety. Leveson and Turner propose extensive education and training of software engineers and more effective communication between manufacturers and their customers.

However, both studies indicate that an understanding of the safety of computer systems must go beyond the laboratory and extend into the organizational settings where it is used. In the case of the Therac-25, it required understanding a complex web of interorganizational

relationships, as well as the technical design and operation of the equipment. Nancy Leveson (1992) points out that most major disasters technological disasters in the last 20 years "involved serious organizational and management deficiencies." Hughes, Randall and Shapiro (1992:119) observe that British no civil collision in UK air space has been attributed to air traffic control failures. But their Mediator control system was failing regularly and had no backup during the period that they studied it. They observe that the reliability of the British air traffic control system resides in totality of the relevant social and technical systems, rather than in a single component.

The need for this kind of organizational understanding is unfortunately slighted in the CS academic world today. *CTF* discusses only those aspects of computer system reliability which are amenable to understanding through laboratory-like studies (Hartmanis and Lin, 1992:110-111). But cases of safety critical systems, like the Therac-25 and British Air Traffic Control, indicate why some Computer Scientists must be willing to undertake (and teach) organizational analysis.

Worldviews and Surprises about Computerization

These few paragraphs barely sketch the highlights of a fertile and significant body of research about computer systems in use. Perhaps the most important simplification for traditional computer scientists is to appreciate how people and their organizations are situated in a social world and consequently compute within a social world. People act in relationship to others in various ways and concerns of belonging, status, resources, and power are often central. The web of people's relationships extend beyond various formally defined group and organizational boundaries (Kling and Scacchi, 1982; Kling, 1987; Kling, 1992). People construct their worlds, including the meanings and uses of information technologies, through their social interactions.

This view is, of course, not new to social scientists. On the other hand, there is no specific body of social theory which can easily be specialized for "the case of computing," and swiftly produce good theories for Organizational Informatics as trivial deductions. The best research in Organizational Informatics draws upon diverse theoretical and methodological approaches within the social sciences with a strong effort to select those which best explain diverse aspects of computerization.

ORGANIZATIONAL INFORMATICS WITHIN COMPUTER SCIENCE

CTF places dual responsibilities on Computer Scientists. One responsibility is to produce a significant body of applicable research. The other responsibility is to educate a significant fraction of CS students to be more effective in conceiving and implementing systems that will enhance organizational performance. It may be possible to organize research and instruction so as to decouple these responsibilities. For example, molecular biologists play only a small role in training doctors. However, CS departments act like an integrated Medical school and Biology department. They are the primary academic locations for training degreed computing specialists, and they conduct a diverse array of less applicable and more applicable research. In practice, the research interests of CS faculty shape the range of topics taught in CS departments, especially the 150 PhD granting departments. CS

curricula mirror major areas of CS research and the topics which CS faculty understand through their own educations and subsequent research. As a consequence, CS courses are likely to avoid important CS topics which appear a bit foreign to the instructor.

An interesting example of this coupling can be illustrated by *CTF*, in a brief description of public-key encryption systems and digital signatures (Hartmanis and Lin, 1992:27). In the simple example, Bob and Alice can send messages reliably if each maintains a secret key. Nothing is said about the social complications of actually keeping keys secret. The practical problems are similar to those of managing passwords, although some operational details differ because the 100 digit keys may be stored on media like magstripe cards rather than paper. In real organizations, people lose or forget their password and can lose the media which store their keys. Also, some passwords can be shared by a group of with shifting membership, and the "secret key" can readily become semi-public. The main point is that the management of keys is a critical element of cryptographic security in practice. But Computer Scientists are prone to teach courses on cryptography as exercises in applied mathematics, such as number theory and Galois theory, and to skirt the vexing practical problems of making encryption a practical organizational activity.

Today, most of the 40,000 people who obtain BS and MS degrees in CS each year in the U.S. have no opportunities for systematic exposure to reliable knowledge about the best design strategies, common uses, effective implementation, and assessments of value of computing in a social world (Lewis, 1989). Yet a substantial fraction of these students go on to work for organizations attempting to produce or maintain systems that improve organizational performance without a good conceptual basis for their work. Consequently, many of them develop systems that underperform in organizational terms even when they are technically refined. They also recommend ineffective implementation procedures and are sometimes even counterproductive.

One defensible alternative to my position is that CS departments should not take on any form of organizational analysis. They should aggressively take a role akin to Biology departments rather than taking on any instructional or research roles like Medical schools. To be sincere, this position requires a high level of restraint by academic Computer Scientists. First and foremost, they should cease from talking about the uses, value or even problems of computerized systems that would be used in any organizational setting. Research proposals would be mute about any conceivable application of research results. Further, they should make effective efforts to insure that anyone who employs their graduates should be aware that they may have no special skills in understanding organizational computing. It would take an aggressive "truth in advertising" campaign to help make it clear that Computer Scientists have no effective methods for understanding computerization in the social world. Further, Computer Scientists would forsake their commitments to subfields like software engineering which tacitly deals with ways to support teams of systems developers to work effectively (Curtis, et. al. 1988). Computer Scientists, in this view, would remove themselves from addressing organizational and human behavior, in the same way that molecular biologists are removed from professionally commenting on the practices of cardiologists and obstetricians. *CTF* argues that this view would be self-defeating. But it would be internally consistent and have a distinctive integrity.

In contrast, CS faculty are often reluctant to wholly embrace Organizational Informatics. But some CS subfields, such as software engineering, depend upon organizational analysis (Curtis, et. al., 1988). Further, CS faculty do little to advertise the distinctive limitations in the analytical skills of our programs' graduates. Part of the dilemma develops because many CS faculty are ambivalent about systematic studies of human behavior. Applied mathematics and other modes of inquiry which seem to yield concise, crisp and concrete results are often the most cherished. As a consequence, those who conduct behaviorally oriented research in CS departments are often inappropriately marginalized. Their students and the discipline suffers as a result.

Between 1986 and 1989, the total number of BS and MS CS degrees awarded annually in the US declined from about 50,000 to approximately 40,000. The number of students majoring in CS rapidly declined at a time when computerization was becoming widespread in many fields. A significant fraction of the decline can be attributed to many students finding CS programs insular and indifferent to many exciting forms of computerization. The decline of military R&D in the U.S. can amplify these trends or stimulate a more cosmopolitan view in CS departments. The decline in military R&D is shifting the job market for new CS graduates towards a markedly more civilian orientation. This shift, along with the trend towards computing distributed into diverse work groups, is leading to more job opportunities for people with CS education who know Organizational Informatics.

The situation of CS departments has some parallels with Statistics departments. Statistics are widely used and taught in many academic disciplines. But Statistics departments have often maintained a monkish isolation from "applications." Consequently, the application of statistics thrives while Statistics departments have few students and modest resources. Might the status of Statistics indicate a future possibility for an insular approach to CS?

The best Organizational Informatics research in North America is conducted by faculty in the Information Systems departments in business schools and by scattered social scientists (cf. Boland and Hirschheim, 1987; Galegher, Kraut and Egidio, 1990; Cotterman and Senn, 1992; Sproull and Kiesler, 1991). But Computer Scientists cannot effectively delegate the research and teaching of Organizational Informatics to business Schools or social science departments.

Like Computer Scientists, faculty in these other disciplines prefer to focus on their own self-defined issues. Computer Scientists are much more likely to ask questions with attention to fine grained technological nuances that influence designs. For example, the professional discussions of computer risks have been best developed through activities sponsored by the ACM's Special Interest Group on Software (SIGSOFT). They are outside the purview of business school faculty and, at best, only a few social scientists are interested in them. Generally, technology plays a minor role in social science theorizing. And when social scientists study technologies, they see a world of possibilities: energy technologies, transportation technologies, communication technologies (including television), medicinal drugs and devices, and so on. They see little reason to give computer-related information technologies a privileged role within this cornucopia. As a consequence, the few social scientists who take a keen interest in studying computerization are unfortunately placed in marginal positions within their own disciplines. Often they must link their studies to

mainstream concerns as defined by the tastemakers of their own fields, and the resulting publications appear irrelevant to Computer Scientists.

Further, faculty in these other disciplines are not organized to effectively teach tens of thousands of CS students, students who are steeped in technology and usually very naive about organizations, about systems development and use in organizations. In North America there is no well developed institutional arrangement for educating students who can effectively take leadership roles in conceptualizing and developing complex organizational computing projects (Lewis, 1989).

CTF is permeated with interesting claims about the social value of recent and emerging computer-based technologies. While many of these observations should rest on an empirically grounded scientific footing, Computer Scientists have deprived themselves of access to such research. For example, the discussion of systems risks in the ACM rests on a large and varied collection of examples and anecdotes. But there is no significant research program to help better understand the conditions under which organizations are more likely to develop systems using the best risk-reducing practices. There is an interesting body of professional lore, but little scholarship to ground it (See Appendix).

Computer Scientists have virtually no scholarship to utilize in understanding when high performance networks, like the National Research and Education Network, will catalyze social value proportional to their costs. Consequently, many of the "obvious" claims about the value of various computing technologies that we Computer Scientists make are more akin to the lore of home remedies for curing illness. Some are valid, others are unfounded speculation. More seriously, the theoretical bases for recommending home medical remedies and new computer technologies can not advance without having sound research programs.

WHAT IS NEEDED

CTF sets the stage for developing Organizational Informatics as a strong subfield within Computer Science. *CTF* bases the expansion of the discipline on a rich array of applications in which many of the effective technologies must be conceived in relationship to plausible uses in order provide attractive social value for multi-billion dollar public investments.

The CS community needs an institutionalized research capability to produce a reliable body of knowledge about the usability and value of computerized systems and the conditions under which computer systems improve organizational performance. In Western Europe there are research projects about Organizational Informatics in a few Computer Science departments and research funding through the EEC's *Espirit* program (Bubenko, 1992; Iivari, 1991; Kyng and Greenbaum, 1991). These new research and instructional programs in Western Europe give Organizational Informatics a significantly more effective place in CS education and research than it now has in North America.

The CS community in the U.S. has 30 years of experience in institutionalizing research programs, especially through the Defense Advanced Research Projects Agency and the National Science Foundation (NSF). There are many approaches, including establishing national centers, supporting individual investigator research grants, supporting short

institutes to help train new investigators and supporting research workshops for ongoing research. All such programs aim to develop and sustain research fields with a combination of direct research funds, the education of future researchers, and the development of research infrastructure. They are all multimillion dollar efforts. Today, NSF devotes about \$125K annually to Organizational Informatics as part of the Information Technology in Organizations program. This start is far short of the level of funding required to develop this field within CS.

The North American CS curricula must also include opportunities for students to learn the most reliable knowledge about the social dimensions of systems development and use (Denning, 1992). These opportunities, formed as courses, can provide varied levels of sophistication. The most elementary courses introduce students to some of the key topics in Organizational Informatics and the limitations of Systems Rationalism as an organizing frame (for example, Dunlop and Kling, 1991a). More advanced courses focus on specific topics, such as those I have listed above. They teach about substantive problems and theoretical approaches for analyzing them. While many of these approaches are anchored in the sociological theory of organizations, CS students usually won't grasp the importance of the theories without numerous computing examples to work with [4]. They also have trouble grasping the character of computing in organizations without guided opportunities for observing and analyzing computerization in practice. Consequently, some courses should offer opportunities for studying issues of computerization in actual organizations.

Fortunately, a few CS departments offer some courses in Organizational Informatics. In addition, some CS faculty who research and teach about human behavior in areas like Human-Computer Interaction and Software Engineering can help expand the range of research and instruction. Curricula would vary, but they should include diverse courses for students who seek basic exposure to Organizational Informatics and those seek more thorough instruction. Unfortunately, only a fraction of the CS departments in the US. have faculty who study and teach about computing and human behavior.

While the study of Organizational Informatics builds upon both the traditional technological foundations of CS and the social sciences, the social sciences at most universities will not develop it as an effective foundational topic for CS. On specific campuses, CS faculty may be able to develop good instructional programs along with colleagues in social sciences or Schools of Management.

But delegating this inquiry to some other discipline does not provide a national scale solution for CS. Other disciplines will not do our important work for us. Mathematics departments may be willing to teach graph theory for CS students, but the analysis of algorithms would be a much weaker field if it could only be carried out within Mathematics Departments. For similar reasons, it is time for academic Computer Science to embrace Organizational Informatics as a key area of research and instruction.

NOTES

[1] See Dunlop and Kling, 1991a for an accessible introduction to these debates. Economic statistics about national level productivity are inexact, and sometimes weak. Baily and Gordon (1988) examined the extent to which measurement problems account for the difficulties of seeing the positive computerization show up in the US national productivity statistics. They concluded that measurements were inexact, and very poor in some sectors like banking, measurement errors were not the primary cause of difficulties.

[2] Organizational Informatics is a new term, and I have found that some people instantly like it while others are put off. I've experimented with alternative labels, like Organizational Computing, which has also resulted in strong and mixed reactions. Computing is a more common term than Informatics, but it's too narrow for some researchers. Informatics also can connote "information," which is an important part of this field. Sociological Computer Science would have the virtues of being a parallel construction of Mathematical Computer Science, but doesn't connote information either. I have not yet found a short distinctive label which characterizes the field and whose connotations are rapidly grasped by both outsiders and insiders.

[3] Jacky's early study was based on published reports, while Leveson and Turner's more thorough study was based upon a significant body of original documents and interviews with some participants.

[4] One hears similar concerns about teaching mathematics to CS students. CS students are much more motivated to learn graph theory, for example, when they learn those aspects which best illuminate issues of computation and when their teaching includes some good computing examples.

REFERENCES

Baily, Martin Neal and Robert J. Gordon. 1988. "The Productivity Slowdown, Measurement Issues, and the Explosion of Computer Power." *Brookings Papers on Economic Activity* 2:347-431.

Bentley, Richard, Tom Rodden, Peter Sawyer, Ian Sommerville, John Hughes, David Randall and Dan Shapiro. 1992. "Ethnographically Informed Systems Design for Air Traffic Control." *Proc. Conference on Computer-Supported Cooperative Work*, Jon Turner and Robert Kraut (ed.) New York, ACM Press.

Boland, Richard and Rudy Hirschhiem (Ed). 1987. *Critical Issues in Information Systems*, New York: John-Wiley.

Bullen, Christine and John Bennett. 1991. "Groupware in Practice: An Interpretation of Work Experience" in Dunlop and Kling 1991b.

Bubenko, Janis. 1992. "On the Evolution of Information Systems Modeling: A Scandinavian Perspective." in Lyytinen and Puuronen, 1992.

Cotterman, William and James Senn (Eds). 1992. *Challenges and Strategies for Research in Systems Development*. New York: John Wiley.

Curtis, Bill, Herb Krasner and Niel Iscoe. 1988. "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM*. 31(11):1268-1287.

Denning, Peter. 1991. "Computing, Applications, and Computational Science." *Communications of the ACM*. (October) 34(10):129-131.

Denning, Peter. 1992. "Educating a New Engineer" *Communications of the ACM*. (December) 35(12):83-97

Dunlop, Charles and Rob Kling, 1991a. "Introduction to the Economic and Organizational Dimensions of Computerization." in Dunlop and Kling, 1991b.

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

Ehn, Pelle. 1991. "The Art and Science of Designing Computer Artifacts." in Dunlop and Kling, 1991.

Fish, Robert S., Robert E. Kraut, Robert W. Root, and Ronald E. Rice. "Video as a Technology for Informal Communication." *Communications of the ACM*,36(1)(January 1993):48- 61.

Galegher, Jolene, Robert Kraut, and Carmen Egido (Ed.) 1990. *Intellectual Teamwork: Social and Intellectual Foundations of Cooperative Work*. Hillsdale, NJ: Lawrence Erlbaum.

Greif, Irene. ed. 1988. *Computer Supported Cooperative Work: A Book of Readings*. San Mateo, Ca: Morgan Kaufman.

Grudin, Jonathan. 1989. "Why Groupware Applications Fail: Problems in Design and Evaluation." *Office: Technology and People*. 4(3):245-264.

Hartmanis, Juris and Herbert Lin (Eds). 1992. *Computing the Future: A Broader Agenda for Computer Science and Engineering*. Washington, DC. National Academy Press. [Briefly summarized in *Communications of the ACM*,35(11) November 1992]

Hewitt, Carl. 1986. "Offices are Open Systems" *ACM Transactions on Office Information Systems*. 4(3)(July):271-287.

Hughes, John A., David Randall, and Dan Shapiro. 1992. "Faltering from Ethnography to Design." *Proc. Conference on Computer-Supported Cooperative Work*, Jon Turner and Robert Kraut (ed.) New York, ACM Press.

Iivari, J. 1991."A Paradigmatic Analysis of Contemporary Schools of IS Development." *European J. Information Systems* 1(4)(Dec): 249-272.

Jacky, Jonathan. 1991. "Safety-Critical Computing: Hazards, Practices, Standards, and Regulation" in Dunlop and Kling 1991b.

Jarvinen, Pertti. 1992. "On Research into the Individual and Computing Systems," in

Lyytinen and Puuronen, 1992.

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

Kling, Rob. 1980. "Social Analyses of Computing: Theoretical Perspectives in Recent Empirical Research." *Computing Surveys*. 12(1):61-110.

Kling, Rob. 1987. "Defining the Boundaries of Computing Across Complex Organizations." *Critical Issues in Information Systems*. edited by Richard Boland and Rudy Hirschheim. pp:307-362. London: John Wiley.

Kling, Rob. 1992. "Behind the Terminal: The Critical Role of Computing Infrastructure In Effective Information Systems' Development and Use." Chapter 10 in *Challenges and Strategies for Research in Systems Development*. edited by William Cotterman and James Senn. Pp. 365-413. New York: John Wiley.

Kling, Rob. 1993. "Computing for Our Future in a Social World" *Communications of the ACM*, 36(2)(February):15-17.

Kling, Rob and Lisa Covi. 1993. Review of *Connections* by Lee Sproull and Sara Kiesler. *The Information Society*, 9(2) (Mar-June).

Kling, Rob and Charles Dunlop. 1993. "Controversies About Computerization and the Character of White Collar Worklife." *The Information Society*. 9(1) (Jan-Feb):1-29.

Kling, Rob, Isaac Scherson, and Jonathan Allen. 1992. "Massively Parallel Computing and Information Capitalism" in *A New Era of Computing*. W. Daniel Hillis and James Bailey (Ed.), pp: 191-241. Cambridge, Ma: The MIT Press.

Kling, Rob and 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.

Kyng, Morton and Joan Greenbaum. 1991. *Design at Work: Cooperative Work of Computer Systems*. Hillsdale, NJ.: Lawrence Erlbaum.

Ladner, Sharyn and Hope Tillman. 1992. "How Special Librarians Really Use the Internet: Summary of Findings and Implications for the Library of the Future" *Canadian Library Journal*, 49(3), 211-216.

Leveson, Nancy G. 1992. "High Pressure Steam Engines and Computer Software." *Proc. International Conference on Software Engineering*, Melbourne, Australia. (May).

Leveson, Nancy G. and Clark S. Turner. 1993. "An Investigation of the Therac-25 Accidents." *Computer July*. (Published in 1992 as Technical Report #92-108. Department of Information and Computer Science, University of California, Irvine.)

Lewis, Philip M. 1989. "Information Systems as an Engineering Discipline." *Communications of the ACM* 32(9)(Sept):1045-1047.

Lucas, Henry C. 1981. *Implementation : the Key to Successful Information Systems*. New York: Columbia University Press.

Lyytinen, Kalle and Seppo Puuronen (Ed.) 1992. *Computing in the Past, Present and Future: Issues and approaches in honor of the 25th anniversary of the Department of Computer Science and Information Systems*. Jyvaskyla Finland, Dept. of CS and IS, University of Jyvaskyla.

Orlikowski, Wanda J. 1993. Learning from Notes: Organizational Issues in Groupware Implementation. *The Information Society* 9(3) (Jul-Sep):237-250. also in Rob Kling. *Computerization and Controversy: Value Conflicts and Social Choices*. (2nd ed.) San Diego, Academic Press, 1996

Poltrock, S.E. and Grudin, J., in press. Organizational Obstacles to Interface Design and Development: Two Participant Observer Studies. *ACM Transactions on Computer and Human Interaction*. (edited version appears in Rob Kling. *Computerization and Controversy: Value Conflicts and Social Choices*. (2nd ed.) San Diego, Academic Press, 1996).

Sarmanto, Auvo. 1992. "Can Research and Education in the Field of Information Sciences Foresee the Future of Development?" in Lyytinen and Puuronen, 1992.

Sproull, Lee and Sara Kiesler. 1991. *Connections: New Ways of Working in the Networked Organization*. Cambridge, Mass.: MIT Press.

Suchman, Lucy. 1983. "Office Procedures as Practical Action: Models of Work and System Design." *ACM Transactions on Office Information Systems*. 1(4)(October):320- 328.

Winograd, Terry and Fernando Flores. 1986. *Understanding Computers and Cognition*. Norwood, NJ: Ablex Publishing.

ACKNOWLEDGEMENTS

This paper builds on ideas which I've developed over the last decade. But they have been deepened by some recent events, such as the *CTF* report. They were also sharpened through a lecture and followon discussion with colleagues at the University of Toronto, including Ron Baeker, Andy Clement, Kelley Gottlieb, and Marilyn Mantei. Rick Weingarten suggested that I write a brief position paper reflecting those ideas. At key points, Peter Denning and Peter Neumann provided helpful encouragement and sage advice. I also appreciate the efforts of numerous other friends and colleagues to help strengthen this paper through their comments and critical assistance. The paper is immeasurably stronger because of the prompt questions and suggestions that I received in response to an evolving manuscript from the following people: Mark Ackerman, Jonathan P. Allen, Bob Anderson, Lisa Covi, Brad Cox, Gordon Davis, Phillip Fites, Simson Garfinkel, Les Gasser, Sy Goodman, Beki Grinter, Jonathan Grudin, Pertti Jarvinen, John King, Heinz Klein, Trond

Knudsen, Kenneth Kraemer, Sharyn Ladner, Nancy Leveson, Lars Matthiesen, Colin Potts, Paul Resnick, Larry Rosenberg, Tim Standish, John Tillquist, Carson Woo and Bill Wulf.

Return to [Rob Kling's](#) homepage, or the [Social Informatics](#) homepage.

Last updated: 1/27/1996

(Indiana update beg. July 1996)