Gaines, B. R. (1988). Intelligent user interfaces. In Expert Systems and Advanced Data Processing. Proceedings of Conference on Expert Systems Technology in the ADP Environment Emrich, M. L., Sadlowe, A. R. and Arrowood, L. F., eds., pp. 271-302. North-Holland, New York.

Invited Address

INTELLIGENT USER INTERFACES

Brian R. Gaines
Knowledge Science Institute
University of Calgary,
Calgary, Alberta, Canada T2N 1N4

ABSTRACT

Requirements for advanced person-computer interaction are analyzed, and the state-of-the-art in dialog design is briefly reviewed. The knowledge structures underlying human-computer interaction as an embedded component of integrated systems are derived. Virtual machine hierarchies for integrated systems and their person-computer interface components are analyzed. Protocol structures are derived for the specification, development and evaluation of intelligent user interfaces.

INTRODUCTION

The person-computer interface still presents major problems in the specification, development and evaluation of computer-based systems. There are three major problems:

- The rapid pace of change in the technology, forms of interaction and system designs.
- Inadequate psychological models of the human operator. There is still a major gap between psychological and cognitive science theories of people, and their behavior in practical situations.
- Inadequate systemic models of computer-based systems. The extreme flexibility of computing technology makes it very difficult to develop general theoretical foundations that have sufficient content to capture critical issues in practical situations.

The growth of opportunities offered by technological developments and applications is counter-balanced by major problems stemming from the pace of change and the lack of foundations:

- The obvious changes are in the *computing technology* where the availability of low-cost graphic displays, graphic input devices, and windowing software has enabled the keyboard and textual interfaces of the 1970s to be replaced, or supplemented, by the window and mouse interfaces of the 1980s. Developments in speech, gesture, and direct sensory interfaces will continue the pace of technical change in the 1990s.
- Other obvious changes are in the *user psychology* where the spectrum of users has extended to include those with no computer experience at one extreme, to those with wide-ranging experience of a variety of past systems at the other.
- There are also major classes of users who are *anomalous* in the experience dimension—the widespread availability of computers in schools has generated a class of users who have experienced many forms of computer interaction but have little or no experience of real-world applications.

Published 1988 by Elsevier Science Publishing Co., Inc.
Expert Systems and Advanced Data Processing
M.L. Emrich, A.R. Sadlowe, and L.F. Arrowood, Editors

There are also major, but less obvious changes occurring in the *types of interaction* occurring between computers and people:

- The step from batch-processing to *interactive use* was a widely recognized transition generating a need for in-depth studies of person-computer interaction and usability.
- The step from episodic, occasional use of computers to *continuous use* of computers as an integral part of the job environment was equally significant, but the change in work patterns and job concepts involved has been less widely studied.
- The step from highly specific computer applications to *integrated systems* having a role in every aspect of one's working life is also leading to major changes in the forms of person-computer interaction and the conceptual structures underlying individual and social activities.
- The increasing complexity of *embedded* interfaces combining special-purpose and general-purpose interactive devices is generating a need to integrate classical ergonomics and person-computer interaction approaches to the interface.

As the technologies, user psychologies and modes of interaction change, so do the theories underlying person-computer interaction and many of the guidelines for effective system design. This is an intrinsic feature of the rapidly changing technology and patterns of use, which is exacerbated by the extreme flexibility of both computers and people.

Computer technology has only a limited grounding in the physical world. Its applications are diverse and may have little inter-relation. Its forms are as much determined by our creative imaginations as they are by its applications. Its applications in themselves may create new infrastructures for social and economic interaction that did not previously exist.

We should expect less coherence and greater volatility in theories and guidelines for person-computer interaction than for any past technology.

INTELLIGENT USER INTERFACES

The emphasis on the problems of person-computer interaction in the last section may make it reasonable to suppose that we should not introduce further major new technologies into the interface. However, there is also the perspective that problems caused by advances in technology may in turn be overcome by further advances in technology, and this has made it reasonable to develop *intelligent user interfaces* in which techniques from artificial intelligence are used to enhance the person-computer interface. Typical developments are in natural language communication and the use of expert system advisers to support system operators.

The use of the term "intelligent" in this context needs some explanation. Computing technology is still far from the general intelligence and common sense of people. However, in well-defined domains it is possible for the system designer to build models of likely transactions and use these to aid and advise the user. If these models are explicit, rather than embedded in code, and sufficiently high-level that they can be understood in non-technical terms, then has become conventional to call the data-structures involved "knowledge." The advantages of explicit, high-level knowledge-driven interfaces are that they can be more easily tailored to specific user communities without reprogramming or deep technical knowledge, and the knowledge structures can be used directly to generate meaningful explanations to users.

Rissland (1984) gives a number of knowledge requirements for such interfaces:-

- Knowledge of the user—for example, preferred style or level of expertise.
- Knowledge of the user's tasks—for example, context or ultimate use of results.
- Knowledge of the tools available and being used—for example, protocols for invocation and use of resources needed.
- Knowledge of the domain of the user's task—for example, epistemological knowledge or measures of importance.
- Knowledge of interaction modalities—for example, how to communicate graphically or by speech.
- Knowledge of how to interact—for example, when to intervene or look over the user's shoulder.
- Evaluation knowledge—for example, how to judge effectiveness or act on gripes.

It is reasonable to call knowledge-based interface designs with some of these characteristics, "intelligent user interfaces", but in studying them more deeply it is best to forget the word "intelligent" and use the availability of overt, high-level knowledge structures as an operational definition. The content of these knowledge structures, and their use to support transactions, then defines the nature and degree of the capabilities of the interface.

AN OVERVIEW OF PAST APPROACHES TO PERSON-COMPUTER INTERACTION

Hansen's (1971) comparison of the interactive system designer with a composer of music appears very apt when one examines the wide variety of styles that have been developed for person-computer interaction. People and computers are very different systems and there are no universally obvious ways in which they should interact. Different designers have different views, often strongly held, and different users have different preferences, often equally strong. Three main styles of dialog may be distinguished (Gaines & Shaw 1984a, Shaw & Gaines 1986b) as shown in Figure 1:

- Formal Dialog: in which the activities and data structures within the computer are presented externally in a direct representation with the minimum syntactic sugar necessary to aid human cognition.
- Natural Language Dialog: in which the human use of language to communicate information and commands is simulated within the narrow context of the activities and data structures within the computer.
- Graphic Dialog: in which the human manipulation of objects within the world to communicate information and commands is simulated to access the activities and data structures within the computer.

Formal Dialog

The prompt-response dialogs of interactive job control languages such as those of VMS, Unix and PC-DOS are typical of formal dialogs. Zloof's (1982) query-by-example database retrieval system is a prime example of the simplicity and power of well-designed formal dialog. It presents the structure of a relational database directly to the user with the minimum syntactic sugar. What the user sees is what is there and the system is reasonably clear to naive users and eminently natural to professional users.

	Graphic Dialog *Representing world*	Formal Dialog *Representing computer*	Natural Language Dialog *Representing person*
2 **1956-63**	**Expensive** Flight simulators & process simulators for limited military and industrial use	**Simplistic** Job control languages giving operators access to machine features	**Output Only** Natural language output from stored script in early computer-aided instruction
3 **1964-71**	**Practical** Mimic diagrams Light pens Touch screens	**Standard** Simple prompt-response Menus Form-filling	**Primitive** Keyword recognition Incorporation of words from input in output
4 **1972-79**	**Creating Reality** Windows Icons Desktop simulation	**Sophisticated** Dialog-engineered prompt-response Interactive form-filling Intelligent form-filling	**Practical** Understanding fixed domain Understanding DBMS domain Meta-level understanding
5 **1980-87**	**Low-Cost** PC flight simulators Integration through Macintosh Integration through Symphony & Framework	**Integrative** Integration through INTELLECT & GURU LISP machine applications Fifth generation objectives	**Sophisticated**

Fig.1 Dialog styles through generations of computing system

Natural Language Dialog

There has always been ambivalence towards natural language dialog because it is remote from the actual operation of current computer systems (Hill 1983). Natural language is attractive because computer users already have the skills to use it to communicate with other people. However, the partial simulations of people that can be programmed currently may be taken by users to have the full capabilities of people because they can "understand" natural language. In sociological terms, it may not be possible to maintain the *role integrity* of a system with a natural language interface. In practice, experience with packages such as Harris' (1984) INTELLECT indicate that they are attractive to users, and problems of role integrity are not significant. It is probably more relevant to contrast applications where a concise formal dialog is adequate and more efficient than natural language with those where requirements are very open-ended and the complex formal language that might be required is best generated by translation from natural language.

Graphic Dialog

Graphic dialog might more generally be termed object simulation since the graphic capabilities of the computer are used to create a model world that mimics the characteristics of a real world natural to the user. The classic examples are the Xerox Star (Smith et al 1983) and, more recently, the Apple Macintosh simulating a desk on which are folders containing documents, in- and out-trays, and a trash bin. As Smith (1983) notes:

> "if everything in a computer system is visible on a display screen, the display becomes reality."

Many industrial and military systems use object simulation as the most natural interface already familiar to the user. Similar design considerations apply as for natural language dialog: graphic dialog is attractive because it brings into play the existing skills of a user, but it becomes very important to ensure that all the user's expectations in that environment are properly fulfilled.

Developmental Sequences Within Dialog Styles

Within each of these styles of interaction there has been a developmental sequence distinguishing levels of capability. Figure 1 tabulates the development of styles of dialog through generations 2 through 5 of computing systems. The center column shows formal dialog representing the computer characteristics directly developing from the early job control languages through: simple prompt-response sequences suited to teleprinters; menus as visual displays became available; form-filling as cursor-addressable displays became available for use in transaction processing; dialog-engineered prompt-response with HELP facilities, default entries, and so on, as user support began to be considered; interactive form-filling as transaction processing became interactive; and intelligent form-filling with fields filled in through calculations as Visicalc was invented at the end of this era.

The left column shows graphic dialog developing from expensive applications in limited military and industrial systems through: mimic diagrams, light pens and touch screens; the windows, icons and artificial reality of Smith's (1977) Pygmalion which led to the Xerox Star; the decline in cost of computer hardware allowing simulation on personal computers; and a variety of integrated systems.

The right column shows natural language dialog developing from the trivial output of stored text in early computer-assisted instruction systems; keyword recognition in later computer-assisted instruction; incorporation of input text in output as in Weizenbaum's (1966) ELIZA and its practical application in Shaw's (1980) PEGASUS; understanding a fixed domain as in Winograd's (1972) SHRDLU; understanding a dynamic database as in Harris' (1977) ROBOT (later INTELLECT); understanding the process of understanding itself as in Davis' TEIRESIAS (Davis and Lenat 1982); and a variety of integrated systems.

Integration of Dialog Styles

Divisions between styles of dialog are beginning to break down as systems become increasingly integrated. The Apple Macintosh integrates graphic dialog in a simulated desk-top environment with the formal dialog of pull-down menus. Packages such as Lotus' Symphony and Ashton-Tate's Framework integrate graphic and formal dialog in a variety of combinations (Gordon 1984). The natural language shells, ASK (Thompson, Thompson & Ho 1983) and INTELLECT (Harris 1984), translate natural user queries into formal DBMS enquiries. The advanced architecture of the Lisp machine (Myers 1982)

supports high-resolution graphics, windows and icons, and is a natural tool for work on all aspects of person-computer interaction including natural language. The fifth generation development program involves the development and integration of all three styles of dialog in increasingly natural person-computer interfaces (Moto-oka 1982).

Dialog Rules for Dialog Styles

As experience in the design and use of interactive systems grew various authors presented guidelines for the design of effective person-computer dialog: Hansen (1971); Wasserman (1973); Martin (1973); Kennedy (1974); Gaines and Facey (1975); Pew and Rollins (1975); Cheriton (1976); Gilb and Weinberg (1977); Gaines (1978); Gebhardt and Stellmacher (1978); Turoff, Whitescarver and Hiltz (1978); Gaines (1981); Gaines & Shaw (1983, 1984a). Shneiderman (1980) has collected many of these rules together, and Maguire (1982) has analyzed some of them systematically, noting contradictions and proposing their resolution. Reference works have also been based on the rules such as the short, practical book of Kidd (1982) and the very detailed reviews of Williges and Williges (1984) and Smith and Mosier (198).

Dialog guidelines are best implemented through an application-independent shell, and Figure 2 gives a hierarchical classification of dialog rules for shell design into: systemic principles; particular features of people and computers; consideration of their interaction; styles or technologies of interaction; and particular applications (Gaines & Shaw 1984b). Such classification is important because of: the proliferation of rules to over 2,000; the development of new technologies for person-computer interaction; the growing maturity of studies of human cognition; person-computer interaction and computer communication protocols; the need for standard dialog engineering tools; and the emergence of commercial products to satisfy that need. However, the design of person-computer interaction in integrated systems needs deeper understanding of the principles underlying communication amongst human and computational entities.

There have been a number of developments that satisfy some of these objectives. Thimbleby (1980) has explored a number of generative principles for dialog design such as dialog determination. Nelson (1980) has proposed virtuality as an analytic principle, noting that we project to the user a virtual world from within the computer. Shneiderman (1983) has subsumed a range of such recent developments in display-orientated systems as examples of direct manipulation. Shackel (1983) has classified them also using the concept of usability. Norman (1984) has defined levels of person-computer interaction. Shaw and Gaines (1984a) have analysed the role of computing as a development in the media of communication. Each of these concepts gives a logic for person-computer interaction design that is supported psychologically and operationally, and hence satisfies some of the objectives expressed above.

The concept of a virtual machine has proved an extremely useful one in computer science (Gaines 1975, Gaines & Shaw 1982, Weegenaar 1978). Its extension from the hardware and software to the programmer and user allows a range of person-computer interaction studies to be set in an integrated framework (Gaines 1979). We can view any computer system as either a programmable machine with certain features together with a program, or as a (programmable) virtual machine with extended features. The user sees the virtual machine not the underlying one, and it is this virtual "reality" that underlies the user interface in machines such as the Star and Macintosh.

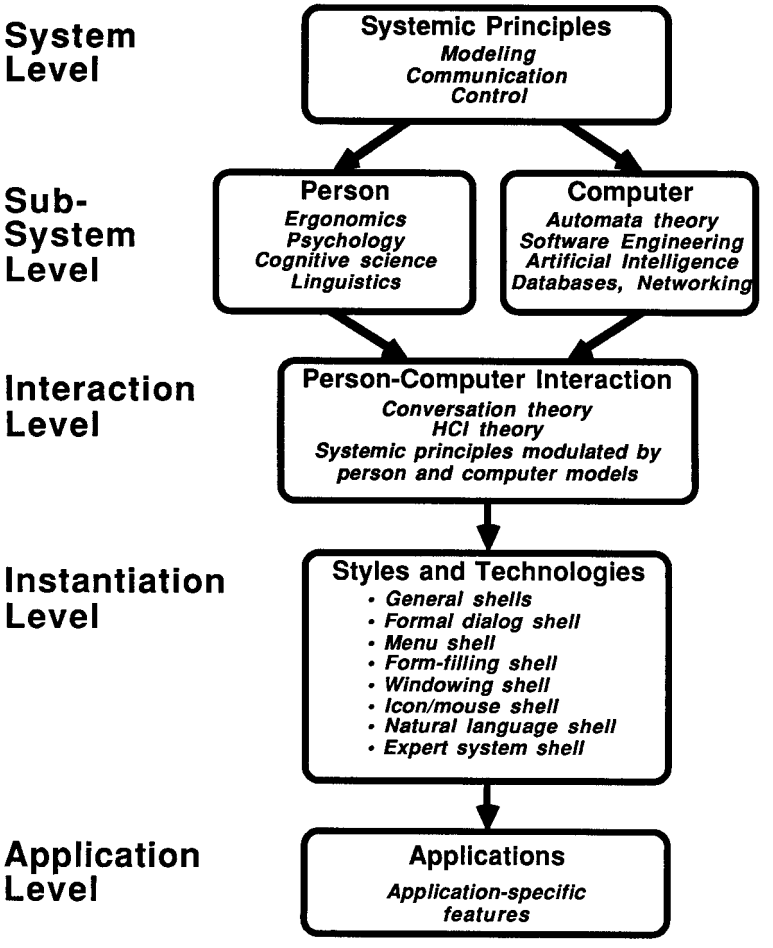

Fig.2 Classification of principles underlying dialog shell design

The algebraic analysis of the virtual machine hierarchy might be extended to include the programmer if we had adequate category-theoretic models of the relevant cognitive system. Some of the formal models being developed in cognitive science show promise of providing this (Weimer & Palermo 1974) such as Scandura's (1973) structural learning. The use of computational models derived from artificial intelligence studies is particularly promising (Bobrow & Collins 1975, Findler 1979, Gentner & Stevens 1983). The explanation of problems of learning structure from behavior in terms of complexity-theoretic measures (Gaines 1976a,b) suggests that they may be valid as a measure of psychological difficulty in understanding virtual machines.

Embedding person-computer interaction within the virtual machine hierarchy gives it a theoretical framework that can both be formalized and linked to many practical consequences and related disciplines. Further research on this approach to the analysis of person-computer interaction problems and principles would be valuable in grounding person-computer interaction within the framework of software engineering. The design principles that we apply to person-computer interaction are part of a larger context of studies of applied psychology, human and computer communication, and general systemic principles. Although they may require fresh validation in the context circumstances of person-computer interaction, they have richer origins and support than studies of person-computer interaction alone. The remainder of the paper analyzes the background to, and foundations for, a deeper understanding of person-computer interaction.

EMBEDDED PERSON-COMPUTER INTERACTION IN INTEGRATED SYSTEMS

In analyzing the knowledge structures underlying person-computer interaction it is important to note that it is not an end in itself but rather part of the operation of integrated systems, and it should be analyzed in a systems context. Figure 3 shows the information systems in a typical organization using integrated computing techniques. The outer ring comprises eight basic information technologies that together support the organization and its interface to the physical and organizational worlds. The center shows the organization itself with its corporate structure, and the rings surrounding it show the human interfaces and communication protocols necessary for the systems' management and integrated operation. The figure has been simplified to the minimum structure necessary to support an effective integrated system, and the residual complexity shows the need to consider the total context of the information technology involved. Such a system will not necessarily succeed if any one component, or even several, are advanced state-of-the-art technologies. However, it will probably fail if one component is inadequate or the total system is not well-integrated.

The Organization

The central core is the non-technology part of the system, the people that manage and operate the plant, their organization and value systems. The overt operation of the integrated system is part of the *technical culture* of the organization in Hall's (1959) terms. It can be communicated, documented and partially programmed. It can also be changed fairly readily in response to new requirements, techniques and technologies. The management structure of the organization is part of its *formal culture*. Only part of it is overt and documented and it is resistant to change. What has come to be called the "corporate culture" (Deal & Kennedy 1982) is in Hall's terms the *informal culture* of the organization. It pervades the organization, is communicated by example, rarely documented, and changes only slowly or under pressures of survival.

The managerial structures and social processes of the organization are not normally treated as part of its information technology. However, in order to be successful the technical systems must be consistent with the organizational structures and processes. This applies particularly to inter-personal communications where computer networks will not be used if they violate the managerial and social norms of the organization, and to command and control paths which have to conform to existing authority and procedures that are not necessarily those of the organization chart.

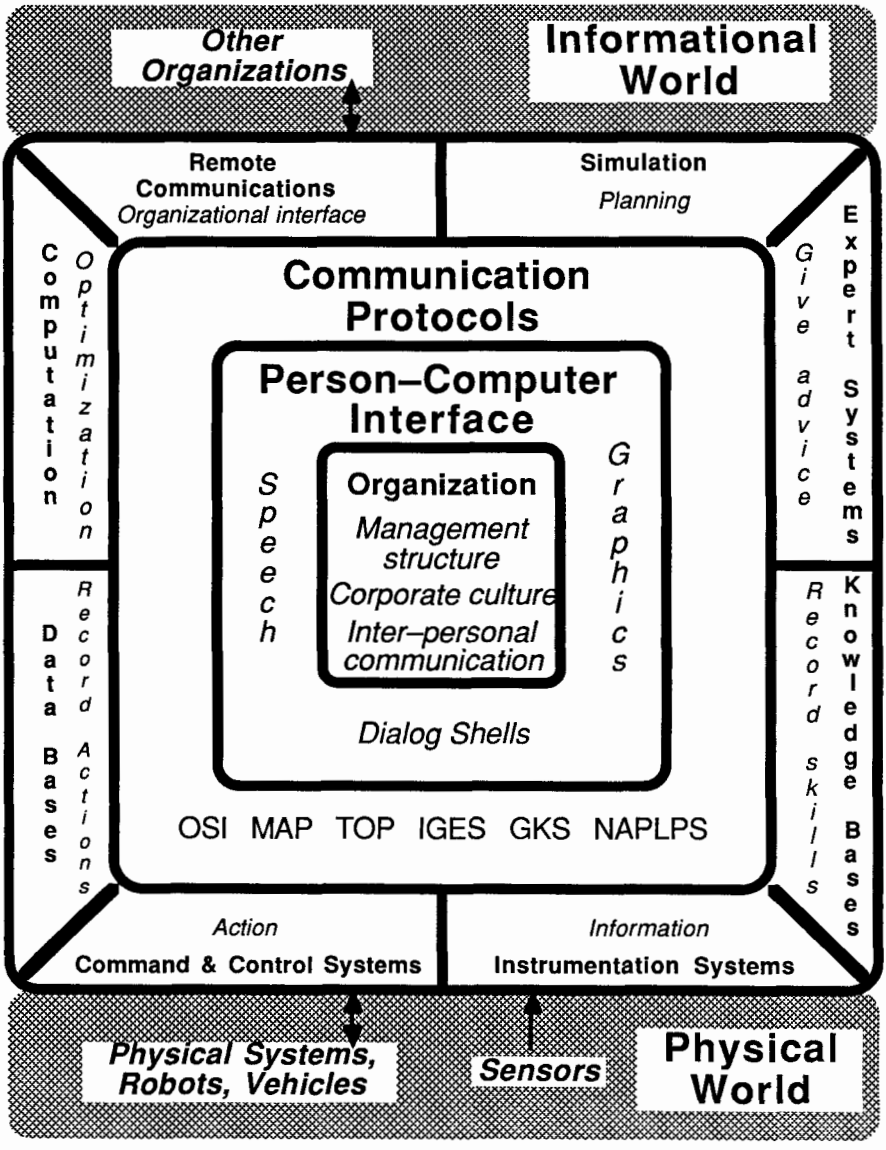

Fig.3 Organizational information systems

Person-Computer Interface

Recognition that the managerial structure of the organization dominates its style of operation draws attention to person-computer interaction as a critical technology. Advances in graphics, voice input/output and intelligent terminals have improved the technical repertoire for interaction. Widespread contact with low-cost personal computers has increased computer literacy and user acceptance of the technology. However, much remains to be done to bring the human engineering of computer-based systems to the same level of professionalism as hardware and software engineering (Gaines & Shaw 1986a,b). For command and control systems involving expert systems and simulation, information presentation and ease of communication are of prime importance. A technically powerful system will be wasted unless it is comprehensible and accessible to its users, and gives them confidence in the basis of its operations.

Communication Protocols

The days of single-vendor, special-purpose systems designed against a static, well-defined specification are long gone. Integrated systems involve all aspects of the operation of a organization. Equipment from many vendors, purchased at different times, for different purposes, by different parts of the organization, has to operate together. In this lies the major practical obstacles to the creation of effective integrated systems at present.

The technical problems are beginning to be resolved at the lower levels by the adoption of industry standards in communication protocols such as the open systems interconnection (OSI) standard (Day & Zimmerman 1983), the provision in data-processing and database packages for the import and export of data from "foreign" systems, and the provision of data-communications for parameter-setting and data-collection in control and instrumentation systems. These alone are necessary but not sufficient to allow the widespread creation of integrated systems. They provide a syntactic framework for system integration but no coherent semantics in which to express it.

The Applications Ring: Basic Technologies

The outer ring in Figure 3 shows the main information technologies in integrated systems and their typical applications. Moving counter-clockwise around the ring:

Remote Communications are necessary to interface the organization with the organizational world in which it operations A particular organization is part of an infrastructure each part of which is trying to optimize its own operations.

Computation Systems perform the classical number-crunching tasks under the control of algorithms that with modern software engineering have a high degree of accuracy, conformity to specification and reliability.

Database Systems have become the classic core of an organization's information system, recording all its activities in a co-ordinated fashion and making the information available on a controlled basis to those systems and people with a legitimate need to know.

Control Systems are now increasingly integrated with information systems so that decisions can be implemented directly in the physical world.

Instrumentation Systems are also increasingly integrated to provide timely information on the state of the physical world and the effectiveness of operations.

The Applications Ring: Decision Technologies

Three technologies in the outer ring of Figure 3 are particularly relevant to decision aids in integrated systems.

Knowledge Bases are distinguished from databases by containing inference rules rather than facts. The rules may necessarily be heuristic in nature, generating plausible advice rather than hard facts. If databases may be said to record an organization's activities then knowledge bases may be said to record its skills.

Expert Systems are the means to apply a knowledge base to the solution of a particular problem. They stand in relation to the knowledge base as does conventional computation to the data base. They differ in that their role as advisors and dependence on heuristics makes *accountability* and *comprehensibility* of major importance. An expert system has to be able to justify and explain the basis of its advice.

Simulation has been separated from other forms of computation because it is computation based on a model of some other system and it is the relation of the simulated model results to that of the system modeled that is of paramount importance. The models used in simulation to infer the behavior of a system are generally deeper than the models used in expert systems to infer the proper advice about a system, even though both may be concerned with the same system.

HIGHER LEVEL PROTOCOLS IN INTEGRATED SYSTEMS

The diagram of Figure 3 captures the key problem areas for integrated systems in a neat picture, but it gives no basis for treating them in an integrated way. How do the corporate culture, person-computer interface and network protocols relate, and how does this affect operations? The only strongly formalized and standardized structures within Figure 1 are the network protocols, and it is useful to examine these in more detail because they involve concepts that are useful in formalizing more of topics encompassed in Figure 1.

Figure 4 shows the Open Systems Interconnection Reference Model. It models communications as a protocol with seven layers each having a distinct function that can be defined fairly independently of the other layers. Within each layer the system can be viewed as providing services to the layer immediately above using the capabilities of the layer immediately below.

The first level defines standards for the physical transfer of messages—electrical levels, timings, connectors, and so on.

The seventh level defines standards for the content of messages in relation to applications programs.

Intermediate levels are concerned with intra-message validity, network addresses, end-to-end message transfer checking, correct sequencing, and standard data formats for very general data structures such as graphics.

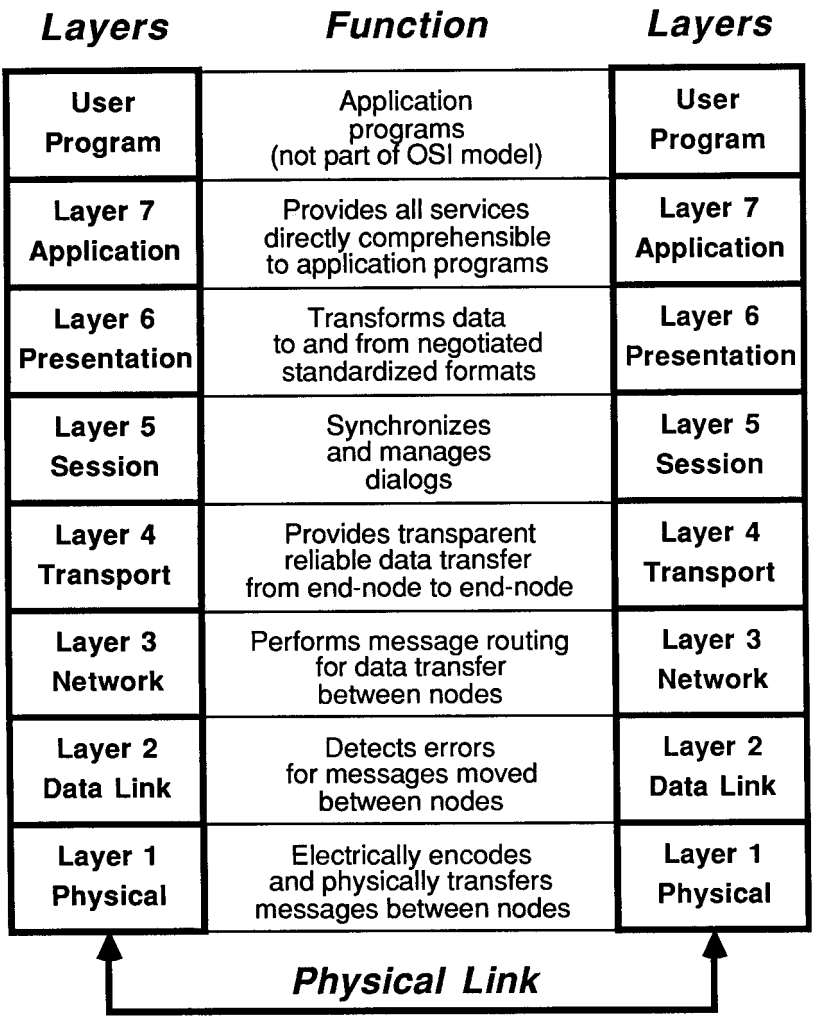

Fig.4 Open systems interconnection reference model

The most important concept underlying the OSI model is shown in Figure 5, that of *virtual circuits*. Designers of a particular level are concerned primarily with its relation with the equivalent level in the system with which it is communicating. They need have little concern with the levels above or below it. Hence they think in terms of messages passing directly between equivalent levels, through a "virtual circuit," rather than through the multiple levels of the actual system.

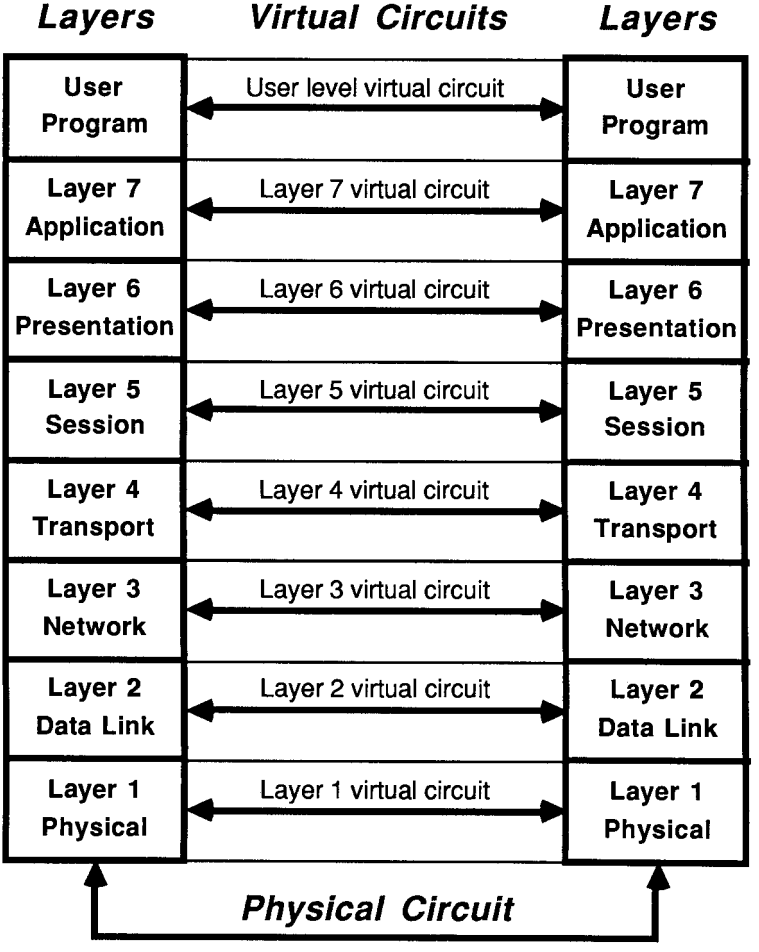

Fig.5 Virtual circuits in the open systems interconnection reference model

The top level above the OSI defined layers in Figure 5 is the "user level virtual circuit," representing the interactions between user programs through the communications network, and hence the interactions between users themselves, and their activities, if the programs are interactive. This concept of virtual circuits between users, and user activities, is very powerful in enabling Figure 3 to be restructured to provide a more integrated framework for the other sections not directly included in the communications protocol. It can be redrawn in such a way that the relation of managerial and operations considerations to higher-level protocols in integrated communications networks is apparent.

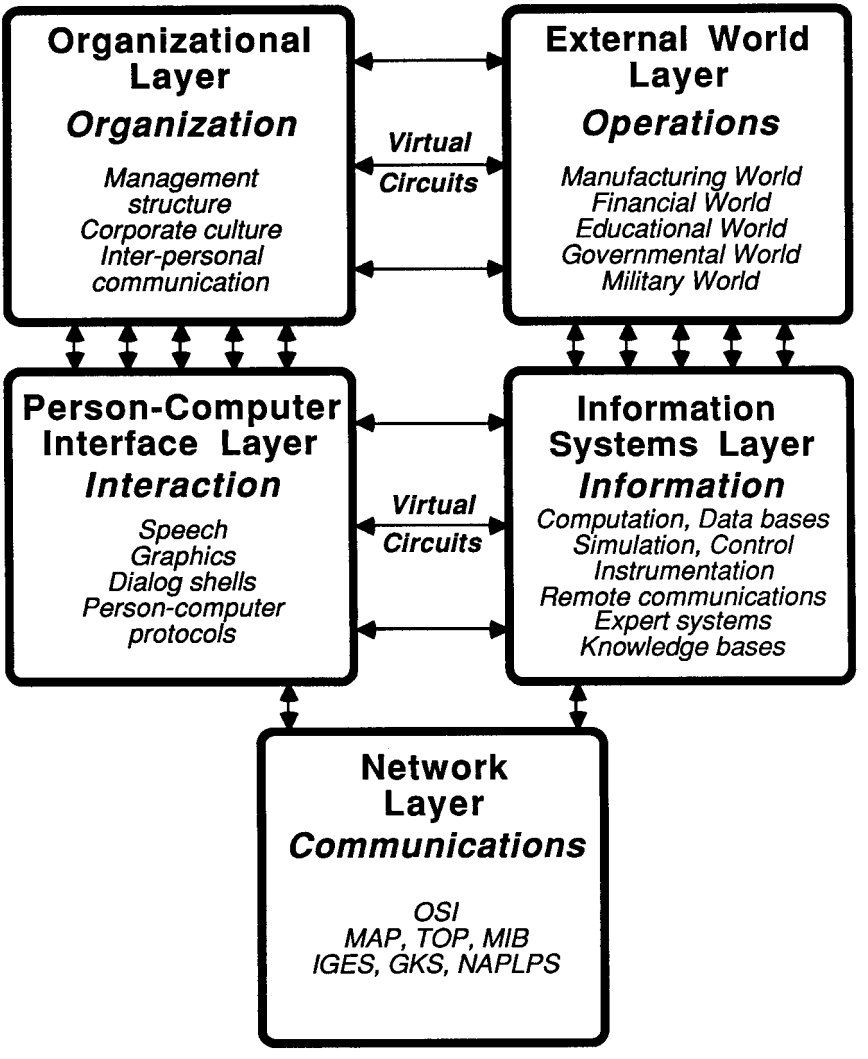

Fig.6 Virtual circuits in organizational information systems

Figure 6 shows the same conceptual blocks as those in Figure 3, but now in hierarchical form with the person-computer interface, information systems, organizational structure and external worlds shown as higher level "layers" above the network layer. In OSI terms these upper layers may be seen as subsuming and extending the presentation and application layers of the protocol, and give substance to the user level virtual circuit.

The virtual circuits between the human-computer interface layer and the information systems layer in Figure 6 are easy to understand. The user of a computer terminal does not think of himself as communicating through a network but rather as directly interacting with some data-processing sub-system. This layer may go beyond the OSI specification but it seems a natural extension of it to the technical aspects of the interaction between people and computing systems. However, the "virtual circuits" at the level above, between the organizational layer, and the external world layer, may seem rather less credible. It is a reasonable metaphor that the sales manager does not see himself as interacting with a complex administrative and productive system, but rather as communicating directly with a customer organization by supplying it with goods. It is a reasonable metaphor that the chief executive does not see himself as concerned with that level of communication but rather as communicating directly with the board and shareholders through a flow of profits and dividends. However, can these metaphors be given technical substance through extension of the layered hierarchical model of virtual communication paths?

The main problem in extending the model to the management level is that people are directly and necessarily involved, and they have different dynamics from technical systems. However, this is also true for the level below where human-computer interaction is involved, and models have been developed at this level that integrate the human and technical components with a hierarchical layered communication model (Gaines 1986). The following section gives systemic foundations for the analysis of integrated systems.

SYSTEMIC FOUNDATIONS FOR INTEGRATED SYSTEMS

In analyzing person-computer interaction it is important to establish what are the critical characteristics of a person and what are those of a computer. How are they differentiated from one another and from other forms of system? Habermas (1981) has emphasized the essential differences between the dynamics of society and that of physical systems. The "laws" that govern human behavior are largely conventions embedded in our society and propagated through our culture (Schutz & Luckman 1973). They do not have the necessity of the laws underlying physical systems and their analysis differs in many ways from those of physical systems (Ulrich 1983).

Human beings create their future through acts of choice that are constrained by their forward-looking intentions. Physical systems unroll their future through acts of necessity that are determined by their previous states. From this point of view, in person-computer interaction we are dealing with phenomena of the life-world which are essentially different from those of the physical world and cannot be encompassed by causal models.

This perspective is radically different from a behaviorist psychological model of people which sees them as conditioned through reinforcement into patterns of behavior that mimic the causal dependencies of physical systems (Skinner 1974, Mackenzie 1977). Habermas' point of view is more akin to a constructivist model of people which sees them as anticipating the future through the development of personal construct systems that are, however, always reconstructable (Kelly 1955). It is reasonable to suppose that people exhibit both these phenomena, and that any model of personal-computer interaction should be able to encompass those aspects of human behavior that are best modeled as causally based and those which are best modeled as anticipatorily based.

The computer component of the system seems straightforward because it has a physical, causal basis. However, computers are programmed by people and hence their behavior also involves human choice. Once they are operating the causal model is appropriate, but in considering system design and person-computer relationships it is important to take the human component into account. The users' mental models may be influenced by their impressions of the designers' intentions. The system designer may be seen as behaving through the computer to anticipate the future behavior of users.

To encompass the possibilities discussed, a conceptual framework is required that can account for these essential differences and similarities, differentiating people from technology and computing from non-programmed physical systems. Popper's (1968) *3 worlds theory* can be used to provide foundations for the distinctions underlying physical processes, human activities, and information technologies. He makes the three-fold distinction: "If we call the world of things—of physical objects—the first world and the world of subjective experience the second world we may call the world of statements in themselves the third world (world 3)." These concepts capture the differences between physical systems and people, and the role of abstraction in moving the distinctions into a third realm regardless of source.

Figure 7 shows the existential hypotheses underlying Popper's 3 worlds as a conceptual framework based on very general distinctions made about distinctions. It introduces a 'World 0' of distinctions in their own right before Popper's criteria are applied. This world is one where distinctions are made without reference to any criteria except that they are useful. It captures the notions of value and of the underlying anarchy that Feyerabend (1975) sees in the scientific process. Gaines (1980) has argued that any distinction generates a system and hence that World 0 as defined here is that of general systems.

- World 0 arises from the assertion that there exist distinctions
  - —this defines a world of *systems*
  - —a key concept is that of *anarchy*, that distinctions may be made freely
  - —the conceptual framework involved is that of *axiology*, that distinctions have value
  - —truth in this world is *utilitarian*, that distinctions made are useful
  - —inference is *pragmatic*, based on chains of reasoning that appear to work
- World 1 arises from the assertion that there exist *physically necessary* distinctions
  - —this defines a world of *physical systems*
  - —a key concept is that of *actuality*, that the distinctions made are actual
  - —the conceptual framework involved is that of *epistemology*, that the distinctions have to be drawn from the actual world
  - —truth in this world is by *correspondence*, that the distinctions made correspond to the properties of physical systems
  - —inference is *causal*, based on chains of physical cause and effect
- World 2 arises from the assertion that there exist *humanly chosen* distinctions
  - —this defines a world of *mental systems*
  - —a key concept is that of *agency*, that distinctions have a maker
  - —the conceptual frameworks involved are those of *psychology and sociology*, that distinctions exist in the mental worlds of individuals and groups
  - —truth in this world is *performative*, that the distinctions chosen will be continue to be chosen
  - —inference is *conventionalist*, based on accepted chains of reasoning that are chosen within a culture

- World 3 arises from the assertion that there exist *formal systems* of distinctions
  - —this defines a world of *coherent systems*
  - —a key concept is that of *abstraction*, that systems of distinctions have their own existence
  - —the conceptual framework involved is that of *ontology*, that some distinctions are made independent of any criteria other than intrinsic structure
  - —truth in this world is by *coherence*, that the systems of distinctions have an acceptable intrinsic structure
  - —inference is *structuralist*, based on chains of reasoning that conform to the structure

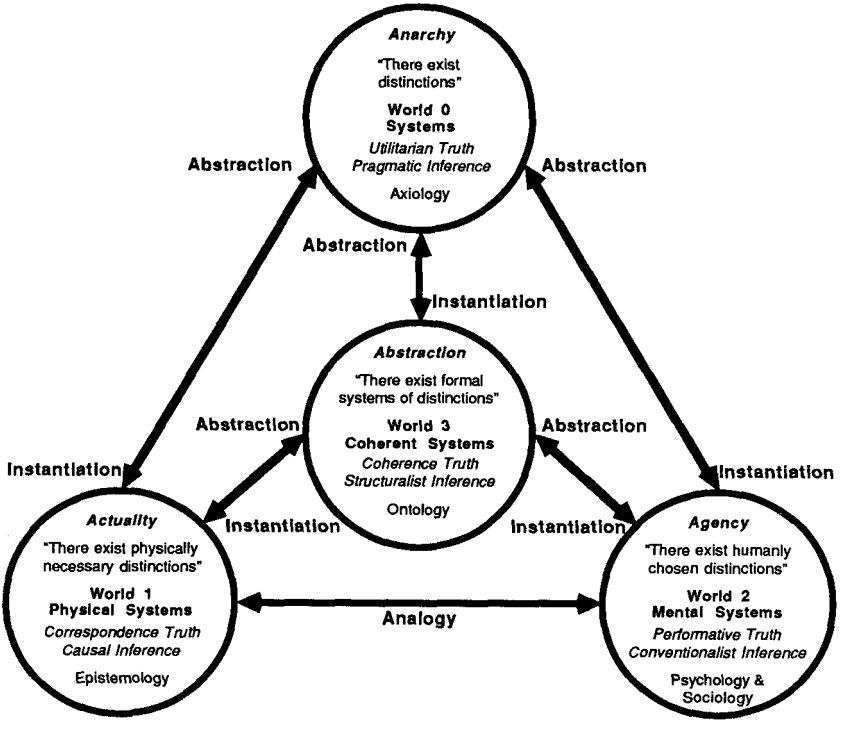

Fig.7 Anarchy, abstraction, actuality, and agency as basic distinctions

Worlds 1, 2 and 3 may be seen as components of World 0 and also as particular instances of it. Similarly Worlds 2 and 3 may be seen as instances of World 1. Conversely World 0 may be seen as an abstraction from Worlds 1, 2 and 3, and World 3 may be seen as an abstraction from Worlds 2 and 3. The conceptual framework of Figure 1 shows how the worlds relate through this *abstraction* and *instantiation*. It also shows the *analogy* between worlds 1 and 2 when we attribute agency to the causal dynamics of physical objects or necessity to the social conventions of human activity.

Figure 2 extends the conceptual framework to show equipment as an instance of a physical system, people as an instance of mental systems, and computers as an instance of both physical and mental systems, inheriting properties through both paths. The peculiar property of electronic digital computers is that they are the archetype of deterministic causal systems, modeled in their behavior as finite-state automata whose next state and output are precisely determined by their current states and inputs. However, they are also programmed devices where the transition tables of the automaton are completely under the control of people—what computers do is what we chose them to do. Hence, they are also the archetype of performative, conventional systems, modeled in their behavior as the intentional artefacts of people, and whose next state and output are in major part determined by open choices made in programming.

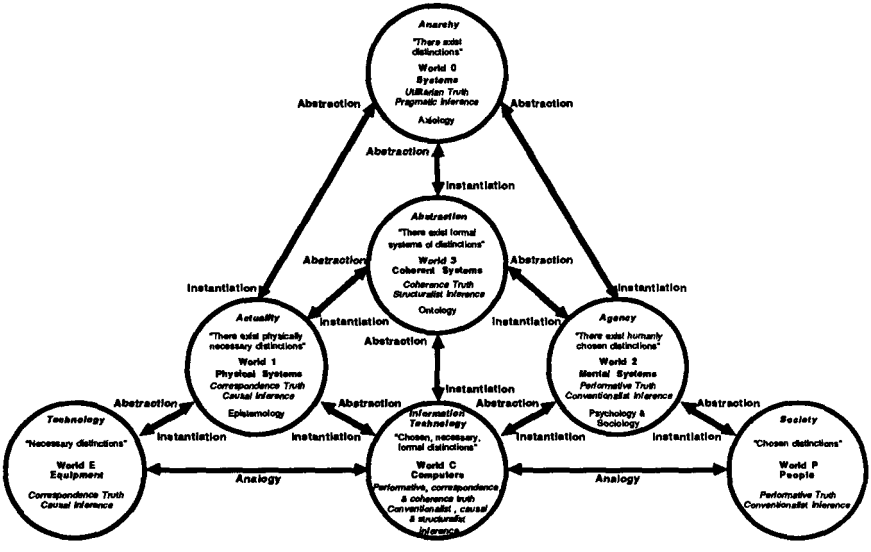

Fig.8 Conceptual structures for person-computer interaction

This dual nature of computing systems underlies many of the problems and controversies in person-computer interaction. The computer is a technological system and it may be advisable to present it to its users as such, avoiding animistic considerations. However, the behavior of the computer is prescribed by people and this underlying prescription may also be evident to the user. The value systems and inter-personal attitudes of the system designer and programmer may have become embedded in the system behavior, making an animistic perspective unavoidable. The performative truth and conventionalist inference aspect of computers is implicit in rules for programming effective person-computer dialog. For example (Gaines & Shaw 1984a), "The system should be consistent in operation—the commands should always do the same thing throughout—the information presentation should always mean the same thing throughout" and "The system should be uniform in operation—the facilities which users have learned to use in one part of the package should be available to them in other parts if they might reasonably expect this", are both example of performative truths allowing inference from reasonable conventions.

The analogies between computers, people and equipment shown in Figure 8 are particularly interesting. If the conventions of the life-world become widely accepted in a culture then they assume the same status as the necessary distinctions of the physical world and the behavior of people within a culture may appear as completely causal. In informal terms, one may be just as hurt attempting to break a moral convention as attempting to walk through a brick wall. In person-computer interaction skilled operators may have highly practiced patterns of behavior that may be modeled through causal dynamics. They are no longer making choices at the lower levels of functioning. This is the basis for some negative "transfer of training" phenomena when functioning developed for one system is carried over to another where it may result in inappropriate choices. All technology has some aspects of human choice embodied in it so that the distinction between technology in general and information technology shown in Figure 8 is not a hard boundary.

The computer is also shown in Figure 8 as an instance of a World 3 system, that is an abstract entity defined in terms of its coherent structure. Category-theoretic models of software systems exemplify the possibility for a high degree of abstraction while still capturing essential features of computing systems (Goguen and Meseguer 1983) and analogies between them, and between them and people (Gaines 1975). The abstract World 3 entity will generally lose some features of both the World 1 physical dynamics that underlies its implementation and the World 2 psychological dynamics that underlies the choice of what to implement.

ANALOGY AND ABSTRACTION IN ANALYZING PERSON-COMPUTER SYSTEMS

The bases for the analogies between computers and people and between computers and other equipment show clearly in Figure 8. The bases of the abstractions that treat people and computers as general systems are also apparent. Figure 9 illustrates how these give rise to a variety of conceptual frameworks for the study of person-computer interaction which have been previously noted as major sources of information about, and insight into, the analysis and design of person-computer systems (Gaines & Shaw 1986b).

System-System Interaction

At the abstract level of system-system interaction there are many system-theoretic principles that are instantiated in the situation where one system is a person and the other a computer. Wiener (1948) emphasized this in his development of cybernetics as the study of "communication and control in men and machines", and many principles of communication theory and control theory apply directly to person-computer interaction. For example, the dialog rule that "all actions are initiated and controlled by the user" (Cheriton 1976) derives from a stability-theoretic result in control theory, that two coupled systems with similar time constants may oscillate unstably around their intended equilibrium state: the person modeling a computer and adapting to it while the computer is modeling the person and adapting to him is a potential source of mutual instability. General systemic principles at the abstraction level interact with more specific considerations at the instantiation level. For example, the dialog rule "avoid acausality" (Gaines 1981) has a system-theoretic foundation in that causal modeling systems generate meaningless models of systems with even slight acausalities. However, to apply this principle we have to know that people are causal modelers (Gaines 1976), and we would not regard the rule as significant unless we knew that time-sharing systems generate apparently random delay distributions.

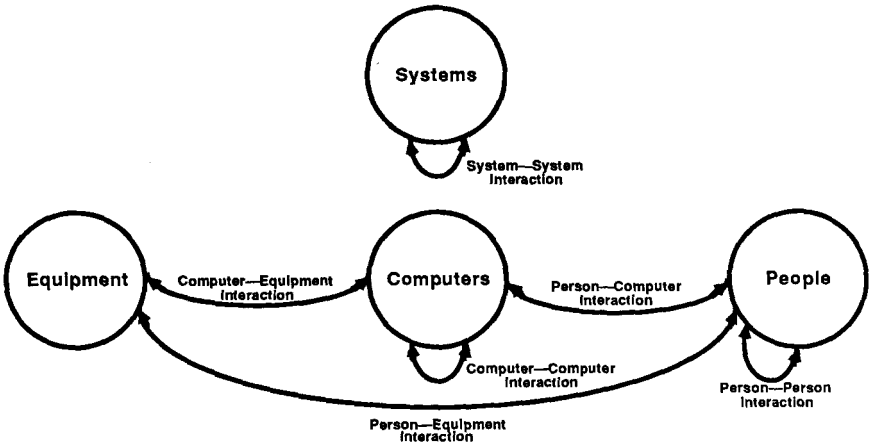

Fig.9 Abstractions and analogies in person-computer interaction

Computer-Equipment Interaction

The analogy from people to physical systems enables person-computer interaction to be seen to be analogous to interfacing a computer to another system such as a piece of equipment. The design principles applicable to computer-equipment interfaces are well known and carry over to person-computer dialog. Problems arise because the system to which the computer is to be interfaced already exists and is not another programmable computer. We may take it as it is and design an interface that copes with its peculiarities. The dialog rule "use the user's model" (Gaines 1981) derives from this, that the dialog engineer should identify the existing interface and attempt to emulate it rather than change it. Problems also commonly arise through noise at the interface and the designer attempts both to provide a low-noise channel and to provide error-detection and correction for unavoidable noise. In person-computer interaction such noise may arise through lack of clarity in information presentation giving rise to perceptual errors in one direction, mis-keying giving rise to errors in the other direction, and so on. The dialog rules "detect user difficulties and assist him in returning to correct dialog" (Cheriton 1976) and "validate data on entry" (Gaines 1981) are principles of communication over a noisy channel.

Computer-Computer Interaction

The analogy from people to computers enables person-computer interaction to be seen to be analogous to computer-computer interaction. It is then possible to define protocols that any programmable system may be reasonably expected to be able to implement. The OSI model previously discussed is particularly relevant because it expresses objectives for computer networks that are equally applicable to people using those networks. The OSI concept is that the network is open to all systems that conform in their communications with certain well-defined protocols. In human terms the protocols may be seen as social norms for the behavior of members of a club; anyone may join provided they agree to conform to these norms. Taylor has applied such protocol concepts to the analysis of person-computer dialog as a hierarchical communications system (Taylor 1987).

Person-Equipment Interaction

The analogy from computers to physical systems enables person-computer interaction to be seen to be analogous to the classic case of man-machine interaction. Consideration of people interacting with equipment has been treated as a branch of applied psychology termed ergonomics that arose, under the same pressures as computer technology, out of World War II studies of pilots, gunners and so on. There is a wide range of results on general problems of human skills, training, its transfer between different learning situations, the effects of fatigue, and so on, that is applicable to person-computer interaction. While interactive computers were used primarily as programming and data entry systems these effects were not major considerations. However, as computer-based interfaces became increasingly the norm for a wide variety of human activities the classic results of applied psychology and ergonomics have become increasingly important. The novelty of the computer should not blind us to commonality with much earlier equipment.

Person-Person Interaction

The analogy from computers to people, enables person-computer interaction to be seen to be analogous to person-person interaction, that is normal linguistic interaction from which the terms human-computer "conversation" and "dialog" in computing terminology have been generalized. Modern linguistic theory (Bennett 1976) has become increasingly concerned with the interaction between participants in a dialog, rather than a view of linguistic output as a predefined stream to be decoded. This provides a rich source of models for person-computer interaction, particularly as artificial intelligence techniques take us closer to emulating people and their language behavior. There are also useful analogies of casual users in transactional analysis of the behavior of strangers meeting (Berne 1974). Taylor notes that many of the principles of language such as scoping and pronomial reference are general principles of communication that occur in gestural graphic dialog with computers that involves no textual interaction (Taylor 1987).

SYSTEM ORIGINS AND ACTIVITY

The principles underlying Figure 9 provide systemic foundations for the forms of reasoning by analogy and abstraction that are commonly used in the analysis and design of person-computer systems. To develop these concepts more formally and provide a framework for the overall dynamics of person-computer interaction, we need to introduce a further distinction, that between the activity and the origins of a system, between its behavior and its structure. In abstract terms the behavior of a system provides a description of what the system does, and the structure of a system provides a description of what the system is. One of the most important problems of system theory is the analysis of the relations between behavior and structure—in one direction, given the structure of a system, to derive its behavior—in the other direction, given the behavior of a system, to derive its structure. In the study of physical systems, mathematical techniques have been developed for moving in both directions with causal models (Gaines 1977, Klir 1985), that is interrelating the necessities of World 1 behavior and structure. However, as already noted, integrated computing systems involve the choice behavior of people and hence show phenomena of the life-world which are essentially different from those of the physical world and cannot be encompassed by causal models.

The dynamics of human behavior are best modelled as those of an *anticipatory system* (Rosen 1985), enhancing its survival by modeling the world, both passively and actively, in order to better anticipate the future. This corresponds to the choice component of World 2 phenomena, that agents are not bound by rigid necessity but can plan and chose certain aspects of their behavior. Figure 10 shows an analysis of the relations between a system, its origins and its activities, when the *actuality—agency* distinction of Figure 1 is also taken into account. The origins of a system have two components: its *causal structure* relating to how it was created; and its *anticipatory structure* relating to why it was created. The activities of a system also have two components: its *causal behavior* relating to how it carries out its activity; and its *anticipatory behavior* relating to why it carries out its activity.

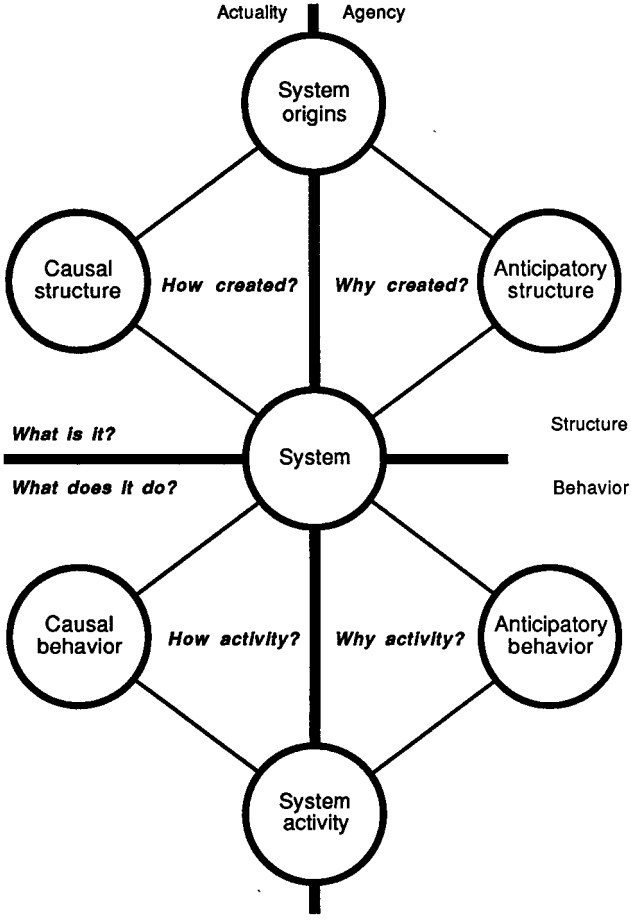

Fig.10 System origins and activity in terms of actuality and agency

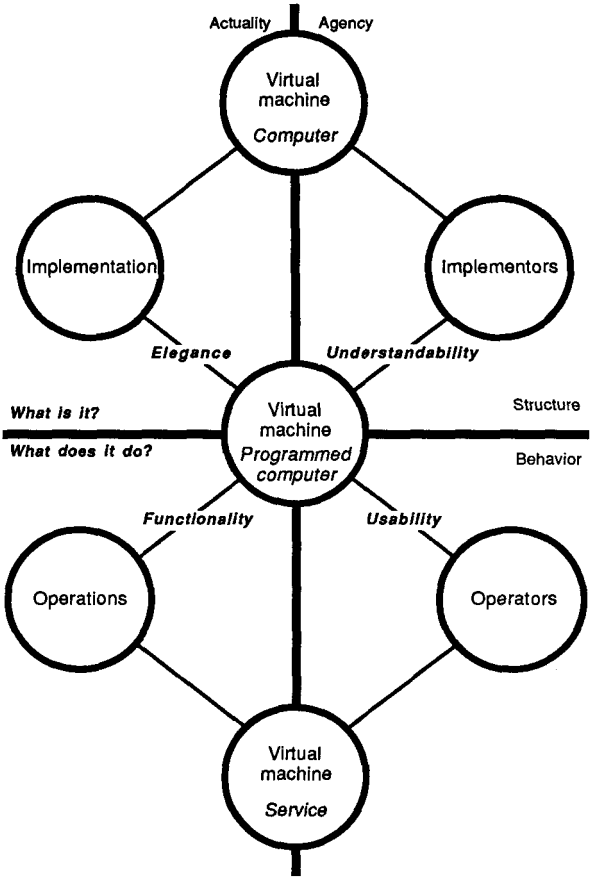

Fig.11 Virtual machines in terms of system origins and activity

The basic distinctions of *actuality—agency* and *structure—behavior* are sufficient to define the virtual machine hierarchy fundamental to computing, and to account for its technical and human components and their relations. Figure 11 fills in Figure 10 to show how the levels of virtual machine arise from the basic distinctions. The origins of any computer-based service are that a *virtual machine*, a computing system, has been programmed by agents, the *implementors*, to create an actuality, the *implementation*. This results in a second virtual machine, the programmed computing system. The activities of the computer-based service are that a *virtual machine*, a programmed computing system, is being operated by agents, the *operators*, to create an actuality, the *operations*. This results in a third virtual machine, the service itself. The relations between the central virtual machine and its structural and behavioral components represent fundamental concepts in system implementation and operation (Edwards and Mason 1987):

- *elegance* considerations, capturing the notion of efficiency in design, concern the relation between the system virtual machine and the underlying resource with which it is implemented—this is another virtual machine capturing the characteristics of the high-level language, operating system, and so on, used in implementation;
- *understandability* considerations, capturing the notion of comprehensibility in design, concern the relation between the system virtual machine and the implementors responsible for creating it;
- *functionality* considerations, capturing the notion of capabilities provided, concern the relation between the system virtual machine and the tasks for which it is being implemented.
- *usability* considerations, capturing the notion of capabilities suitably interfaced, concern the relation between the system virtual machine and the operators who use it.

The logic underlying Figures 10 and 11 may be iterated to interpolate as much detail as required between the upper and lower levels—what is "activity" at one level becomes "structure" for the lower level. Figure 12 shows an analysis of the key features of a computer system in terms of six layers:

- At the top level the overall computer system originates in terms of purpose and structure as part of the culture within which it embedded. This *cultural layer* captures the milieu within which the system has been generated and can itself be subject to detailed analysis.
- At the next level the activities generated through this structure and purpose lead to a system of intentionality in which anticipatory activity leads to the acquisition of knowledge, or models of the world, and the generation of goals.
- At the next level the activities generated through anticipation lead to actions and communications—a linguistic distinction between activities directed to World 1 effects and World 2 effects, respectively. This *knowledge* layer captures the internal processes supporting the modeling and control activities of an anticipatory system.
- At the next level the activities generated through action and communication have to be transmitted to some external medium. This *action* layer captures the internal processes supporting the interfacing of an anticipatory system to the world.
- At the next level the activities generated through transmission have to be expressed in such a way as to have the desired effect in the world. This *expression* layer captures the internal processes supporting the encoding of actions and communications.
- At the lowest level the activities generated through expression exist physically in the external world. This *physical* layer captures the external effects of the activities of an anticipatory system.

It is interesting to relate the iterated levels of Figure 12 back to the terminology of Figure 11. First, consider Figure 12 as a conceptual framework for a person. The culture is the 'implementor' of the person, taking the genetic 'virtual machine' and socializing it to have intentions and knowledge that conform with the purpose of a society. The intentions of the person are carried out by the formation of anticipatory systems that acquire knowledge and generate goals. These in turn implement goal-directed actors or communicators, which themselves implement processes to transmit the activities of the actors to the world, ultimately as sequences of acts that are directly expressed in the physical world. Second, consider Figure 12 as a conceptual framework for a computer system, an intelligent integrated system. The diagram from bottom up represents the evolution of computer technology, from the physical device technology to carry out acts, through the programmed control of action, through the knowledge-based derivation of action, through the goal-creating activities of intentionally-specified systems, to the origins of information technology in our culture at the top level.

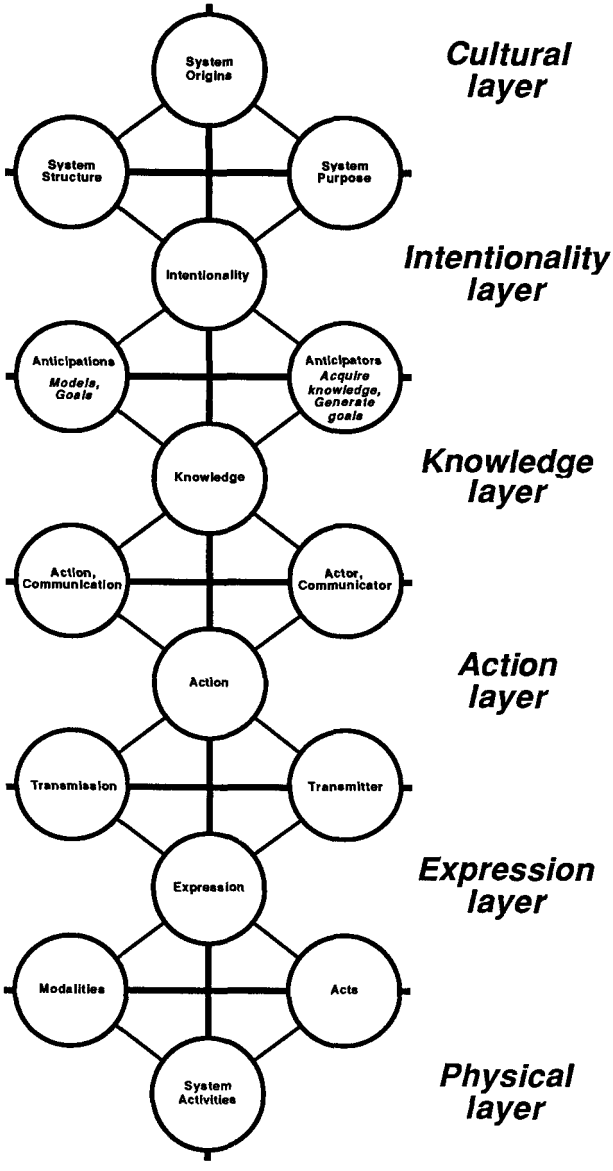

Fig.12 The hierarchy of layers in a computing system

Figure 13 extends Figure 12 to multiple systems showing the virtual circuits in, and between, two people communicating through a computer system. What is particularly interesting about this diagram is that the same distinctions, terminology and model are being applied to the people, their interactions with each other, their interface to the information technology, and its interface to other information technology. The same systems principles apply to the psychology, sociology, human-computer interaction, and computer-computer interaction because computing systems have the dual identity shown in Figure 8. They are both technological and humanistic systems and it is human component, the choice available in programming, that determines their roles and behavior in interacting with people.

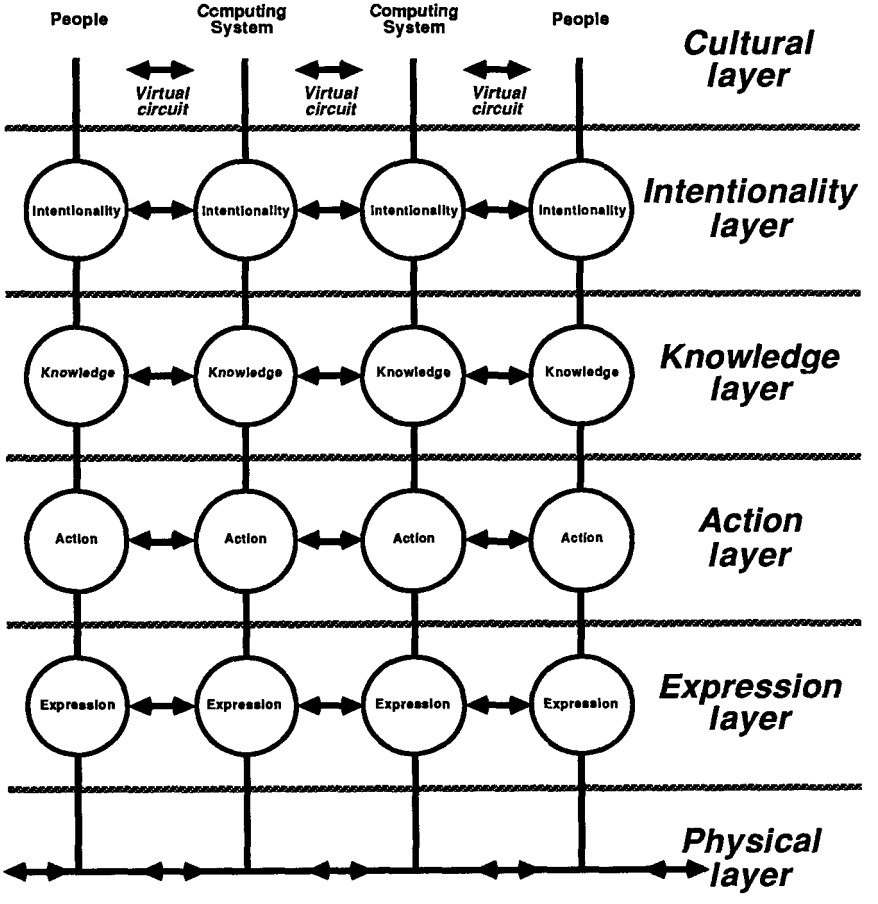

Fig.13 Virtual circuits between layers in computing systems

FROM DESCRIPTION TO PRESCRIPTION

The conceptual framework for person-computer interaction developed above is a descriptive knowledge structure showing what distinctions a system designer might be expected to make and how these derive from fundamental systemic principles. However, critical distinctions are the basis of the major prescriptive design rules that the designer has to keep in mind when making significant decisions. If the distinction is important to the design then it must be maintained or obviated and this requires two activities, each of which leads to design criteria which may be in opposition. First, the distinction may be considered to be significant to maintain and this leads to design considerations based on its maximizing its importance. Second, the distinction may be considered exaggerated and this leads to design considerations based on minimizing its importance. This possibility for opposition is what can lead to contradictory guidelines which are both significant to take into account (Maguire 1982).

For example, maintaining or the distinctions between layers in Figures 12 and 13 may lead to opposing design considerations. One group of designers may wish to maintain the importance of the intentionality layer and note that the intentions of the human participants in using the system are important. The users do not see themselves as interacting with interesting technology but as carrying out an organizational function. The sales manager does not see himself as interacting with a complex administrative and productive system, but rather as communicating directly with a customer organization by supplying it with goods. The chief executive does not see himself as concerned with that level of communication but rather as communicating directly with the board and shareholders through a flow of profits and dividends. The system must be designed around the perspectives of its users in terms of their intentions. However, another group of designers may wish to obviate this distinction and be concerned with users only in terms of their actions noting, perhaps, that in their situation the intentions of users are variable and not available to the system. It would be confusing for the system to make assumptions about, or attempt to infer, the user's intentions. It should be designed around the perspectives of users in terms of their actions.

Thus conceptual structures are not only descriptive, providing a vocabulary and semantic relationships for talking about interface design. They are also prescriptive in identifying the critical dimensions along which design decisions must be analyzed. At an abstract level a conceptual structure is arbitrary like the grid lines on a map serving a useful structure in dividing up a territory so that different parts of it may be discussed. However, the concrete conceptual structure of an expert is not arbitrary but more like the contour lines of map in following natural phenomena and significant boundaries. The distinctions made are significant in that they indicate differences where decisions are most likely to be relevant to the problems encountered.

CONCLUSIONS

Requirements for advanced person-computer interaction have been analyzed, and the state-of-the-art in dialog design briefly reviewed. The knowledge structures underlying human-computer interaction as an embedded component of integrated systems have been derived. Virtual machine hierarchies for integrated systems and their person-computer interface components have been analyzed. Protocol structures have been derived for the specification, development and evaluation of intelligent user interfaces.

Figure 14 summarizes the features of the theoretical model. It extends the layered communications model of computer communications to incorporate the interactions between people and information systems, and between the organization and the business world. It distinguishes the physical world of material flows from the psycho-social world of knowledge and intentionality flows, and the logical world of information and control flows. Provided these basic distinctions are made, however, it is possible to model the complete structure and activities encompassed in Figure 14 within a unified conceptual framework. Intelligent user interfaces incorporate parts of this framework as explicit knowledge structures.

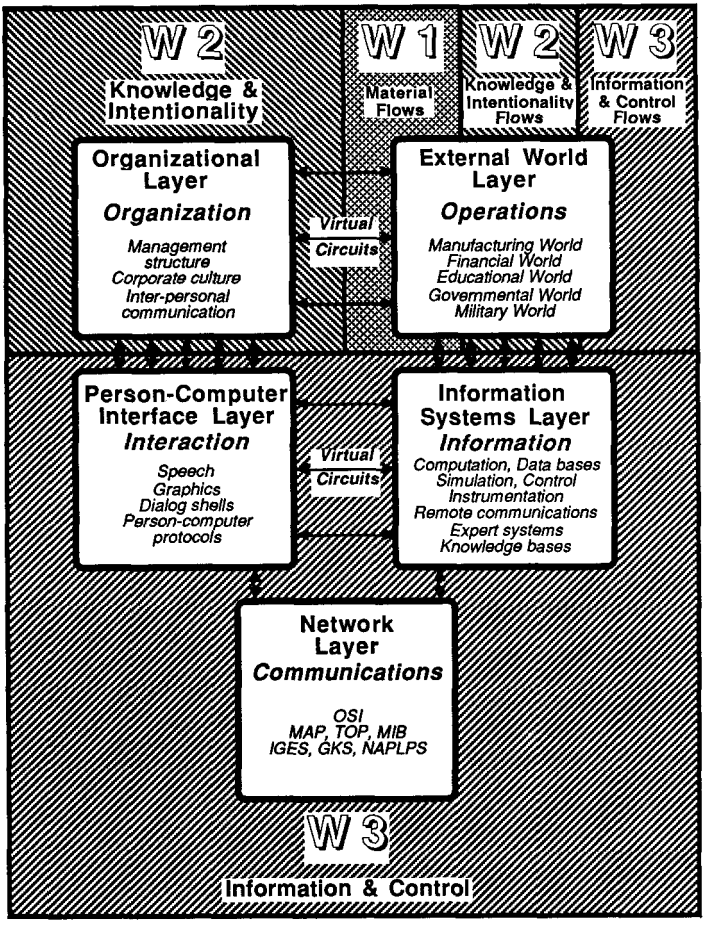

Fig.14 Material, knowledge & intentionality, and information & control flows

ACKNOWLEDGEMENTS

This paper has been influenced by discussions over many years and I am grateful to Walter Doherty, Jack Edwards, Peter Facey, Saul Greenberg, Martin McGuire, Mildred Shaw, Ben Shneiderman, Martin Taylor, Harold Thimbleby, Ian Witten and various anonymous referees. Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Bennett, J. (1976). **Linguistic Behaviour**. Cambridge, UK: Cambridge University Press.

Berne, E. (1974). **What Do You Do After You Say Hello?**. London: Andre Deutsch.

Bobrow, D.G. & Collins, A., Eds. (1975). **Representation and Understanding: Studies in Cognitive Science**. New York: Academic Press.

Cheriton, D.R. (1976). Man-machine interface design for time-sharing systems. **Proceedings of the ACM National Conference**, 362-380.

Davis, R. & Lenat, D.B. (1982). **Knowledge-Based Systems in Artificial Intelligence**. New York: McGraw-Hill.

Day, J.D. & Zimmerman, H. (1983). The OSI reference model. **Proceedings IEEE**, **71**(12), 1334-1340 (December).

Deal, T.E. & Kennedy, A.A. (1982). **Corporate Cultures: The Rites and Rituals of Corporate Life**. Reading, Massachusetts: Addison-Wesley.

Edwards, J.L. & Mason, J.A. (1986). Evaluating the intelligence in dialogue systems. **International Journal of Man-Machine Studies, 27**, to appear.

Feyerabend, P. (1975). **Against Method**. London: NLB.

Findler, N., Ed. (1979). **Associative Networks: Representation of Knowledge by Computers**. New York: Academic Press.

Gaines, B.R. (1975). Analogy categories, virtual machines and structured programming. Goos, G. & Hartmanis, J., Eds. **GI—5.Jahrestagung. Lecture Notes in Computer Science, 34**, 691-699. Berlin: Springer Verlag.

Gaines, B.R. (1976a). On the complexity of causal models. **IEEE Transactions on Systems, Man & Cybernetics, SMC-6**(1), 56-59 (January).

Gaines, B.R. (1976b). Behaviour/structure transformations under uncertainty. **International Journal of Man-Machine Studies, 8**(3), 337-365 (May).

Gaines, B.R. (1977). System identification, approximation and complexity. **International Journal of General Systems, 3**, 145-174.

Gaines, B.R. (1978). Programming interactive dialogue. **Pragmatic Programming and Sensible Software**, 305-320. Uxbridge, Middlesex, UK: Online Conferences.

Gaines, B.R. (1979). The role of the behavioural sciences in programming. **Structured Software Development Vol.2**. pp. 57-68. Maidenhead, UK: Infotech International.

Gaines, B.R. (1980). General systems research: quo vadis? Gaines, B.R., Ed. **General Systems 1979**. Vol. 24, pp. 1-9. Kentucky: Society for General Systems Research.

Gaines, B.R. (1981). The technology of interaction—dialogue programming rules. **International Journal of Man-Machine Studies, 14**(1), 133-150 (January).

Gaines, B.R. (1986). Cognitive models of human-computer interaction in distributed systems. **Proceedings of 1986 International Conference on Systems, Man and Cybernetics, IEEE 86CH2364-8**. pp.1358-1362. Atlanta (October).

Gaines, B.R. & Facey, P.V. (1975). Some experience in interactive system development and application. **Proceedings IEEE, 63**, 155-169 (June).

Gaines, B.R. & Shaw, M.L.G. (1982). Analysing analogy. Trappl, R., Ricciardi, L. & Pask, G., Eds. **Progress in Cybernetics and Systems Research Vol.IX**. pp. 379-386. Washington: Hemisphere.

Gaines, B.R. & Shaw, M.L.G. (1983). Dialog engineering. Sime, M. & Coombs, M.J., Eds. **Designing for Human-Computer Interaction**. pp. 23-53. London: Academic Press.

Gaines, B.R. & Shaw, M.L.G. (1984a). **The Art of Computer Conversation: A New Medium for Communication**. New Jersey: Prentice Hall.

Gaines, B.R. & Shaw, M.L.G. (1984b). Dialog shell design. **Proceedings of INTERACT'84: 1st IFIP Conference on Human-Computer Interaction**, 344-349. Amsterdam: North-Holland.

Gaines, B.R. & Shaw, M.L.G. (1986a). From timesharing to the sixth generation: the development of human-computer interaction part I. **International Journal of Man-Machine Studies, 24**(1), 1-27 (January).

Gaines, B.R. & Shaw, M.L.G. (1986b). Foundations of dialog engineering: the development of human-computer interaction part II. **International Journal of Man-Machine Studies, 24**(2), 101-123 (February).

Gebhardt, F. & Stellmacher, I. (1978). Design criteria for documentation retrieval languages. **Journal of the American Society for Information Science, 29**(4), 191-199 (July).

Gentner, D. & Stevens, A.L., Eds. (1983). **Mental Models**. Hillsdale, New Jersey: Lawrence Erlbaum.

Gilb, T. & Weinberg, G.M. (1977). **Humanized Input**. Cambridge, Mass., USA: Winthrop Publishers.

Goguen, J.A. & Meseguer, J. (1983). Programming with parametrized abstract objects in OBJ. **Theory and Practice of Programming Technology**. Amsterdam: North-Holland.

Gordon, P. (1984). What is integrated software?. **PC World, 2**(11), 72-77 (October).

Habermas, J. (1981). **The Theory of Communicative Action: Reason and the Rationalization of Society**. Boston: Beacon Press.

Hall, E.T. (1959). **The Silent Language**. New York: Doubleday.

Hansen, W.J. (1971). User engineering principles for interactive systems. **Proceedings of the Fall Joint Computer Conference, 39**, 523-532. New Jersey: AFIPS Press.

Harris, L.R. (1977). User oriented data base query with the ROBOT natural language query system. **IJMMS, 9**(6), 697-713 (November).

Harris, L.R. (1984). Experience with INTELLECT. **AI Magazine, 5**(2), 43-50.

Hill, I.D. (1983). Natural language versus computer language. Sime, M. & Coombs, M.J., Eds. **Designing for Human-Computer Interaction**. pp. 55-72. London: Academic Press.

Kelly, G.A. (1955). **The Psychology of Personal Constructs**. New York: Norton.

Kennedy, T.C.S. (1974). The design of interactive procedures for man-machine communication. **International Journal of Man-Machine Studies, 6**, 309-334.

Kidd, A. (1982). **Man-Machine Dialogue Design**. BTRL, Ipswich, UK: Martlesham Consultancy Services.

Klir, G.J. (1985). **Architecture of Systems Problem Solving**, New York: Plenum Press.

Mackenzie, B.D. (1977). **Beahviourism and the Limits of Scientific Method**. London: Routledge and Kegan Paul.

Maguire, M. (1982). An evaluation of published recommendations on the design of man-computer dialogues. **International Journal of Man-Machine Studies, 16**(3), 237-261 (April).

Martin, J. (1973). **Design of Man-Computer Dialogues**. New Jersey, USA: Prentice-Hall.

Moto-oka, T., Ed. (1982). **Fifth Generation Computer Systems**. Amsterdam: North-Holland.

Myers, W. (1982). Lisp machines displayed at AI conference. **Computer, 15**(11), 79-82 (November).

Nelson, T. (1980). Interactive systems and the design of virtuality. **Creative Computing, 6**(11), 56-62 (November).

Norman, D.A. (1984). Stages and levels in human-machine interaction. **IJMMS, 21**(3) (September).

Pew, R.W. & Rollins, A.M. (1975). Dialog specification procedure. **Report No.3129**. Cambridge, Massachusetts: Bolt, Beranek & Newman.

Popper, K.R. (1968). Epistemology without a knowing subject. Van Rootselaar, B., Ed. **Logic, Methodology and Philosophy of Science III**. pp. 333-373. Amsterdam, Holland: North-Holland Publishing Co..

Rissland, E.L. (1984). Ingredients of intelligent user interfaces. **IJMMS, 21**(3) (September).

Rosen, R. (1985). **Anticipatory Systems**. Oxford: Pergamon.

Scandura, J.M. (1973). **Structural Learning. I. Theory and Research**. New York: Gordon & Breach.

Schutz, A. & Luckmann, T. (1973). **The Structures of the Life-World**. London: Heinemann.

Shackel, B. (1983). Usability: its meaning and evaluation in human-computer systems. **Tutorial 9 at CHI'83**. Boston: ACM SIGCHI.

Shaw, M.L.G. (1980). **On Becoming a Personal Scientist**. London: Academic Press.

Shneiderman, B. (1980). **Softare Psychology**. Cambridge, Massachusetts: Winthrop.

Shneiderman, B. (1983). Direct manipulation: a step beyond programming languages. **Computer, 16**(8), 57-69 (August).

Skinner, B.F. (1974). **About Behaviourism**. London: Jonathan Cape.

Smith, D.C. (1977). **Pygmalion**. Basel: Birkhauser.

Smith, D.C. (1983). Visibility in the Star user interface. **Proceedings of Graphics Interface '83**, 181. Ottawa: National Research Council of Canada.

Smith, D.C., Irby, C., Kimball, R., Verplank, B. & Harslem, E. (1983). Designing the Star user interface. Degano, P. & Sandewall, E., Eds. **Integrated Interactive Computing Systems**. pp. 297-313. Amsterdam: North-Holland.

Smith, S.L. & Mosier, J.N. (1986). **Guidelines for Designing User Interface Software**. ESD-TR-86-278. Bedford, Massachusetts: Mitre Corporation (August).

Taylor, M.M. (1987). Layered protocols for computer-human dialogue: I Principles. **International Journal of Man-Machine Studies, 27**, to appear.

Taylor, M.M., McCann, C.A. & Tuori, M.I. (1984). The interactive spatial information system. **DCIEM Report 84-R-22**. Toronto: DCIEM.

Thimbleby, H. (1980). Dialogue determination. **International Journal of Man-Machine Studies, 13**(3), 295-304 (October).

Thompson, B.H., Thompson, F.B. & Ho, T.-P. (1983). Knowledgeable contexts for user interaction. **Proceedings of National Computer Conference, 52**, 29-38. Arlington, Virginia: AFIPS Press.

Turoff, M., Whitescarver, J. & Hiltz, S.R. (1978). The human machine interface in a computerized conferencing environment. **Proceedings of the IEE Conference on Interactive Systems, Man and Cybernetics**, 145-157.

Ulrich, W. (1983). **Critical Heuristics of Social Planning**. Bern: Haupt.

Wasserman, T. (1973). The design of idiot-proof interactive systems. **Proceedings of the National Computer Conference, 42**, M34-M38. New Jersey: AFIPS Press.

Weegenaar, H.J. (1978). Virtuality and other things like that. **Proceedings of IEEE COMCON, CH1388-8/78**, 287-293.

Weimer, W.B. & Palermo, D.S., Eds. (1974). **Cognition and the Symbolic Process**. Hillsdale, New Jersey: Lawrence Erlbaum.

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. **Journal of the ACM, 9**, 36-45.

Wiener, N. (1948). **Cybernetics**. Cambridge,Massachusetts: MIT Press.

Williges, B.H. & Williges, R.C. (1984). Dialogue design considerations for interactive computer systems. Muckler, F.A., Ed. **Human Factors Review: 1984**. pp. 167-208. Santa Monica, California: Human Factors Society.

Winograd, T. (1972). **Understanding Natural Language**. Edinburgh: Edinburgh University Press.

Zloof, M.M. (1982). Office-by-example: a business language that unifies data and word processing and electronic mail. **IBM Systems Journal, 21**(3), 272-304.