# KNOWLEDGE STRUCTURES FOR INTELLIGENT INTERACTION

Brian R. Gaines

Knowledge Science Institute, University of Calgary
Calgary, Alberta, CANADA T2N 1N4

## ABSTRACT

This paper presents systemic foundations for complex systems of computers and people operating together to pursue common goals. A general systems analysis is developed based on the knowledge structures underlying the distinctions between technology and people, and between computers and non-programmed technology. This is used to show how various forms of analogy and abstraction may be used to derive design principles for person-computer interaction. The systemic analysis is extended to include relations between system structure and behavior, and used to develop a hierarchical model of the protocols in complex person-computer systems.

## INTRODUCTION

We are now in the fifth generation computing era with its emphasis on complex, knowledge-based systems involving close person-computer interaction [13,21].The multi-task, multi-user, multi-modal systems of the fourth and fifth generations go beyond human factors knowledge based on the textual display and keyboard technology of yesterday. If we continue to rely on empirical knowledge based on studies of the past we shall never be equipped to deal with the systems of the present let alone those of the future.

New technologies create new problems but they also often provide new means to solve them. The development of knowledge-based systems has focused attention on techniques for *knowledge engineering*, the acquisition of knowledge from human experts and its transfer to expert systems [14]. Effective techniques for knowledge engineering are based on the observation that there are common patterns in the way in which we model the world, explain phenomena, anticipate events and communicate knowledge. These patterns are part of the process of human understanding and may be termed general systems principles [16]. Their significance is that, since they underlie physical laws and social conventions, their identification in particular modeling schema enables us to analyze those schema in a universal framework.

The extraction of these systemic principles is straightforward if we examine the systems of *distinctions* being used in the evaluation and protocol methodologies. As Brown has noted, by analyzing the distinctions made in any discipline, "we can begin to reconstruct, with an accuracy and coverage that appear almost uncanny, the basic forms underlying linguistic, mathematical, physical and biological science, and can begin to see how the familiar laws of our own experience follow inexorably from the original act of severance" [3].

## ABSTRACTION AND ANALOGY

When people face new situations where empirical data and models are lacking, there are various patterns of reasoning that they use. One is argument by analogy based on relating the new situation to known situations of different types but similar structures. Another is argument by abstraction based on discarding specific features of the situation and analyzing in terms of basic laws and systemic principles. As Lakoff and Johnson have noted the linguistic metaphors resulting from these analogies and abstractions can become so much a part of our thinking and vocabulary that we are no longer aware of their metaphorical nature [20]. The terminology of computing abounds with words that were originally specific to human knowledge processes but are now used for equipment. We forget the origins of these words and the metaphorical thinking from which they derive.

Analogy and abstraction and both be misleading if, for example, similarity in partial structure does not extend to further structure or key elements of the situation are lost through abstraction. In analyzing person-computer interaction it is important to establish what are the

Copyright 1987 by Elsevier Science Publishing Co., Inc.
Methodologies for Intelligent Systems
Z. W. Ras and M. Zemankova, Editors

critical characteristics of a person and what are those of a computer. How are the differentiated from one another and from other forms of system? Habermas has emphasized the essential differences between the dynamics of society and that of physical systems [18].

The "laws" that govern human behavior are largely conventions embedded in our society and propagated through our culture. They do not have the necessity of the laws underlying physical systems and their analysis differs in many ways from those of physical systems. 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.

In person-computer interaction we are dealing with phenomena of the life-world [25] which are essentially different from those of the physical world and cannot be encompassed by causal models [28]. We are also dealing with phenomena of computing which have a physical, causal basis. However, we have to remember that the computing was programmed by people and hence also involves human choice within the framework of the life-world. We need a systemic model that accounts for these essential distinctions, differentiating people from technology and computing from non-programmed physical systems, while supporting the analogies and abstractions that we use in analyzing and designing person-computer systems.

Popper's *3 worlds theory* [22] can be used to provide basic systemic foundations for the distinctions underlying the physical processes, human activities, and information technologies. He bases his theory on Bolzano's notion of "truths in themselves" in contradistinction to "those thought processes by which a man may grasp truths", proposing that "thoughts in the sense of contents or statements in themselves and thoughts in the sense of thought processes belong to two entirely different worlds" [24], and making 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 1 shows the existential hypotheses underlying Popper's 3 worlds as a knowledge structure 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 sees in the scientific process [6]. Gaines has argued that this any distinction generates a system and hence that World 0 as defined here is that of general systems [10].

- World 0 arises from the hypothesis 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 hypothesis that there exist *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 hypothesis that there exist *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 hypothesis that there exist distinctions *independent* of their source
  - —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

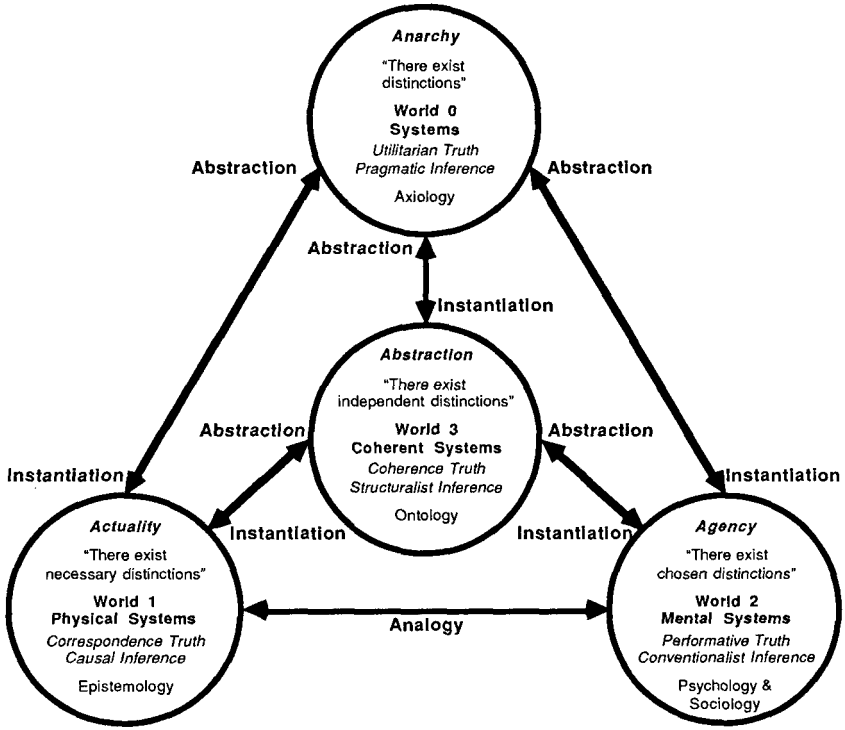


**Fig.1 Anarchy, abstraction, actuality, and agency as systemic distinctions**

The knowledge structure of Figure 1 shows how the the worlds relate through *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 knowledge structure 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.

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

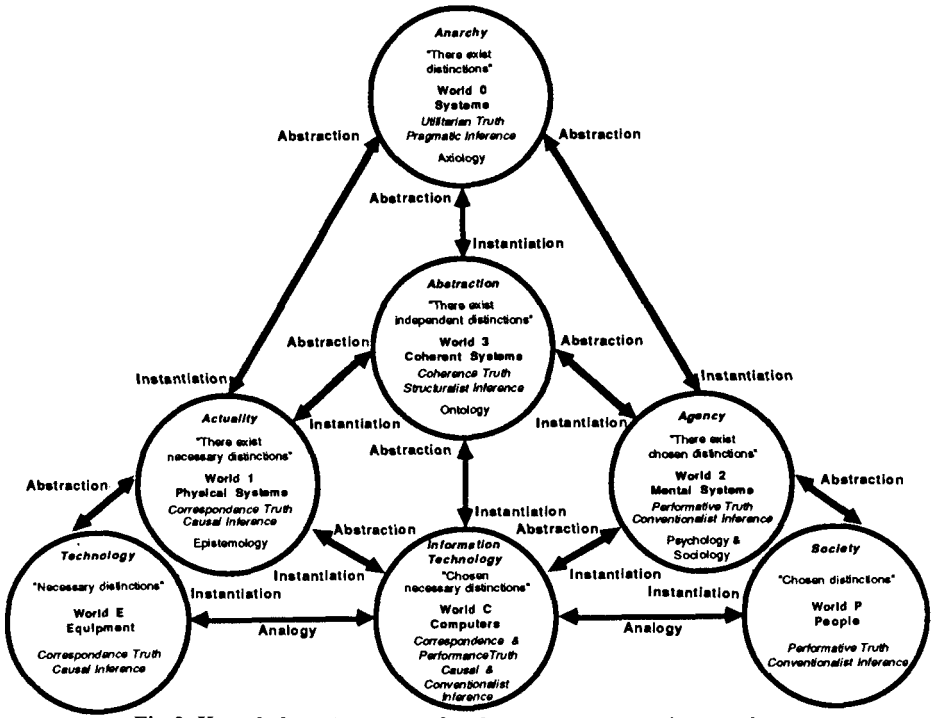


**Fig.2 Knowledge structures for human-computer interaction**

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 very evident in rules for programming effective person-computer dialog. For example, "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" [15], are both example of performative truths allowing inference from reasonable conventions.

The analogies between computers, people and equipment shown in Figure 2 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 the skilled operator may have highly practiced patterns of behavior that may be modeled through causal dynamics. He is no longer making choices at the lower levels of functioning. This is the basis for "transfer of training" phenomena when automatic functioning developed for one system is carried over to another where it may result in inappropriate choices. It should also be noted that 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 2 is not a hard boundary.

The computer is also shown in Figure 2 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 [17]. Such models also gives foundations for analogical relations between computing systems, and also for the person-computer analogy [7].

## KNOWLEDGE STRUCTURES FOR EQUIPMENT, COMPUTERS & PEOPLE

The bases for the analogies between computers and people and between computers and other technical equipment show up clearly in Figure 2. The bases of the abstractions that treat both computers and people as general systems are also apparent. Figure 3 illustrates how these analogies and abstractions give rise to a variety of models 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.

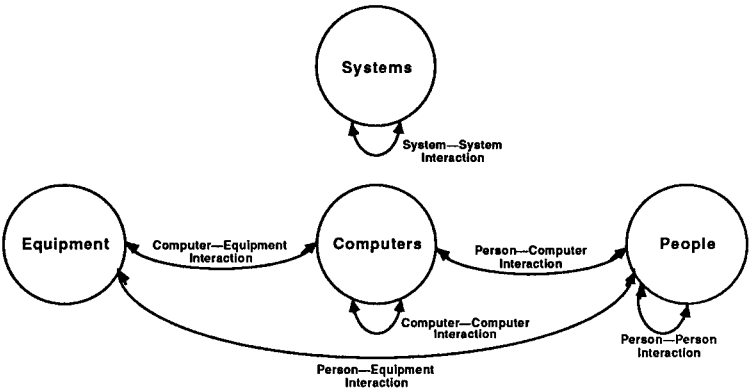


**Fig.3 Abstraction and analogies in the analysis of person-computer interaction**

### 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 emphasized this in his development of cybernetics as the study of "communication and control in men and machines" [29], and many principles of communication theory and control theory apply directly to person-computer interaction. For example, the dialog rule that "the user should dominate the computer" [11] 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" [11] 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 [8], and we would not regard the rule as significant unless we knew that time-sharing system generate apparently random delay distributions.

### 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 to take it as it is and design an interface that copes with its peculiarities. The dialog rule "use the user's model" [11] 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 rule "validate data on entry" [11] is a principle 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 it is reasonable to expect any programmable system to be able to implement. The Open System Interconnection standard [4] is particularly interesting because it hierarchically structures computer-computer protocols for networks in a way that may have relevance for person-computer protocols. For example, dialog engineering rules may be seen as defining a "human protocol" [26]. The concept of an open system is itself relevant because it expresses objectives for computer networks that are equally applicable to people using those networks. The aim is to allow integrated systems to be formed from multiple components not all from one vendor and not all installed at the same time [12]. 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 provide 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 [27].

### 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 wealth of results on general problems of human skills, training, its transfer between different learning situations, the effects of fatigue, and so on, that is immediately 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. We do not have the time and effort to waste on rediscovering what is already known.

### 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 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 [1]. 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, for example "What do you say after you say hello?" [2]. Taylor notes that many of the principles of language such as scoping and pronomial reference are general principles of communication that occur in graphic/gesture dialog with computers that involves no textual interaction [27].

## SYSTEM ORIGINS AND ACTIVITY

The principles underlying Figure 3 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 participant systems, 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 [9,19], 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* [23], 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 4 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.

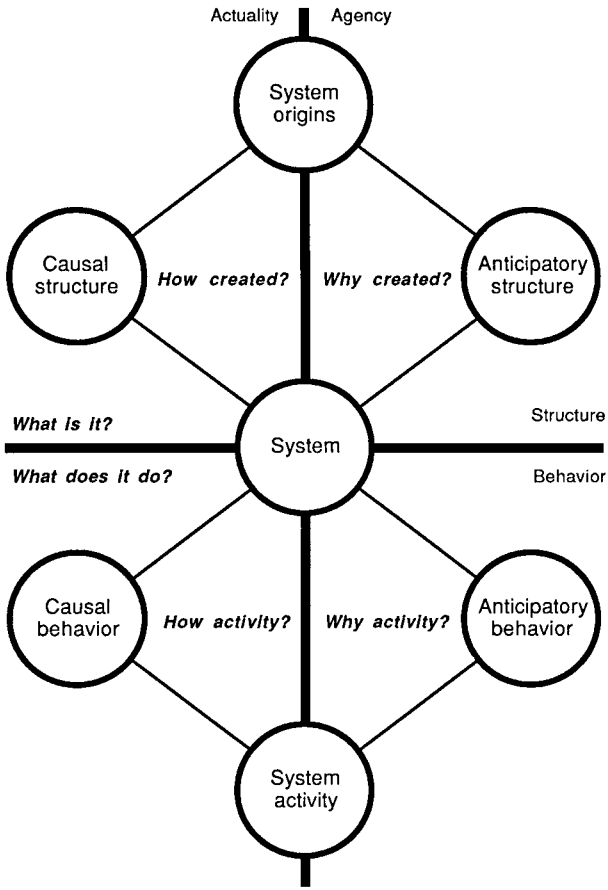


**Fig.4 System origins and activity in terms of actuality and agency**

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 5 fills in Figure 4 to show how the levels of virtual machine arise from the basic distinctions.

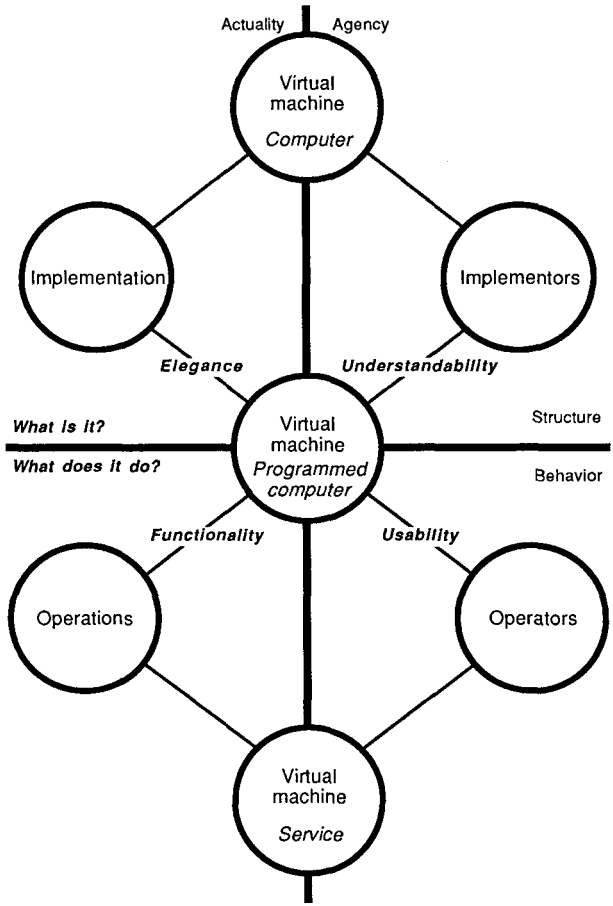


**Fig.5 Virtual machines in terms of system origins and activity**

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 [5]:

- *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 who created it;
- *functionality* considerations, capturing the notion of capabilities provided, concern the relation between the system virtual machine and the tasks for with 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.

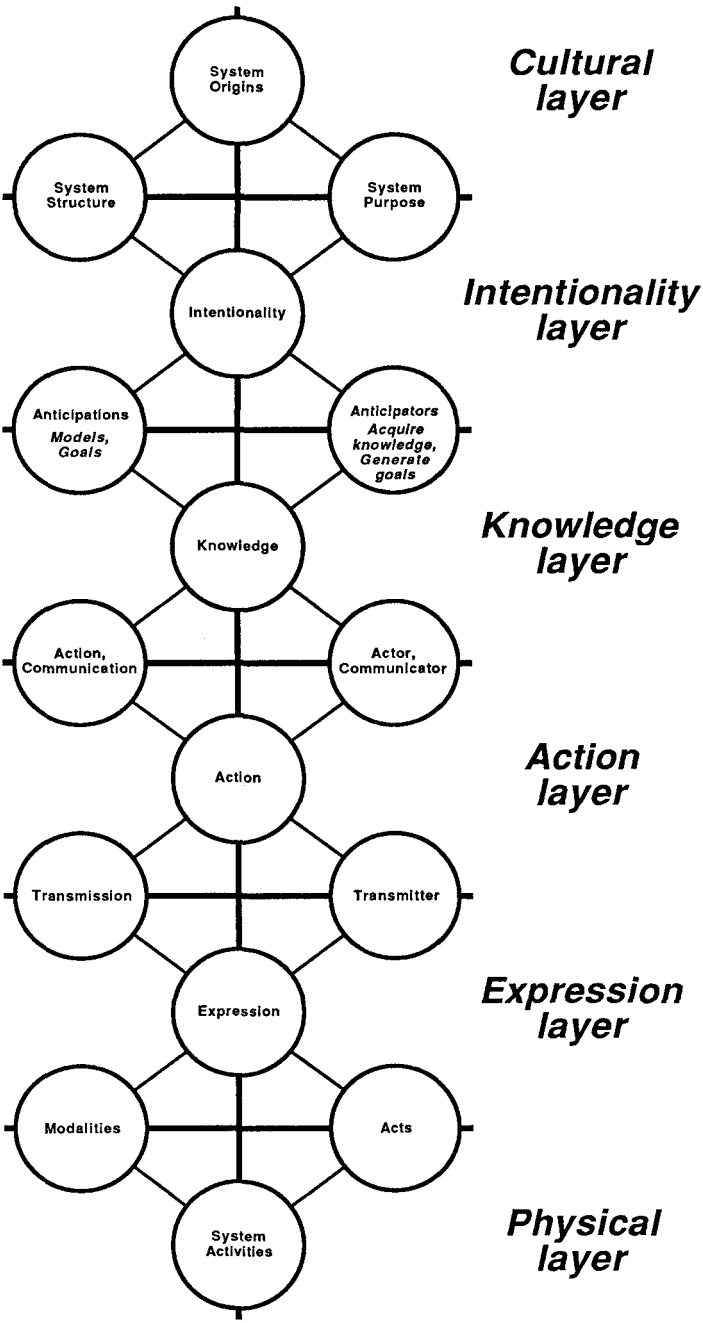


Fig.6 The hierarchy of layers in a computing system

The logic underlying Figures 4 and 5 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 6 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. This *intentionality* layer captures the anticipatory nature of an intelligent system in supporting its survival through prediction and action.
- 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 6 back to the terminology of Figure 5:

- First, consider Figure 6 as a model of a person. Systemically, 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 particular 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 6 as a model of 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 in fifth-generation systems, through the goal-creating activities of intentionally-specified systems in future-generation computing, to the origins of information technology in our culture at the top level.

Figure 7 extends Figure 6 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 6. 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.

## CONCLUSIONS

Complex interactive systems are being built and, despite many problems, they will be made to work because they are needed to deal with some of the very complex activities necessary to our society. Conventional empirical studies of human-computer interaction based on single users operating at standard workstations do not, and cannot, give adequate foundations for the design and analysis of participant systems. We have to go back to the systemic foundations of people, computing and person-computer interaction and develop new models that can encompass the problems and design considerations of complex interactive systems. The analysis developed in this paper is foundational and yet does provide an adequate framework within which to analyze the structure and operation of experimental participant systems developed to date.

Establishing knowledge structures for the computer and human components of complex systems provides an operational framework within which to express specifications, designs, implementations and evaluations. The knowledge structures can be simply and naturally expressed within an object-orientated expert system shell and then applied to design, decision-making and simulation. Specific systems become represented as particular instantiations of the general knowledge structure and analogical relations between systems can be formally expressed as inheritance rules in the knowledge hierarchy. The knowledge structures are available in overt form, yet they are not frozen but can be simply updated. The consequences of changes in the structures can be immediately evaluated in terms of their impact of past case histories and future designs. The development of complex intelligent integrated systems requires the use of intelligent tools with the maximum power available. Well-founded knowledge structures for human-computer interaction provide us with computer-aided design and computer-aided manufacturing tools for complex systems.

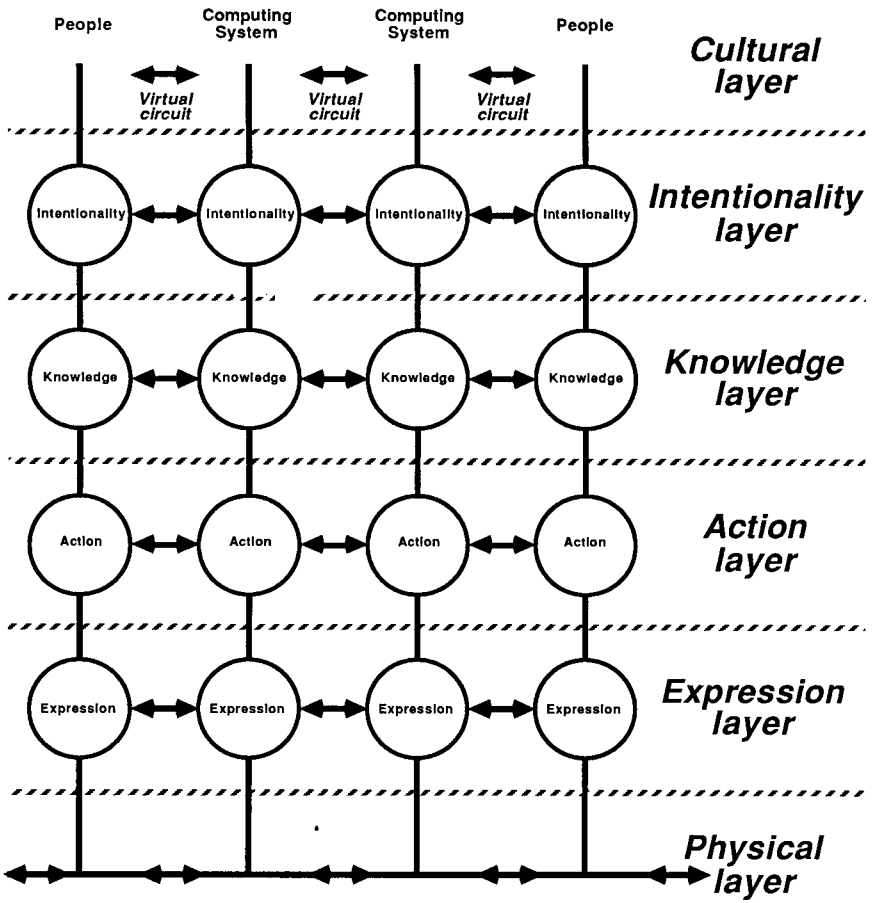


**Fig.7 Virtual circuits between layers in computing systems**

## ACKNOWLEDGEMENTS

Financial assistance for the studies on which it is based has been made available by the National Sciences and Engineering Research Council of Canada.

## REFERENCES


1. Bennett, J. **Linguistic Behaviour**. Cambridge, (Cambridge University Press 1976).
2. Berne, E. **What Do You Do After You Say Hello?**. (Andre Deutsch, London 1974).
3. Brown, G.S. **Laws of Form**. (George Allen & Unwin, London 1969).
4. Day, J.D. & Zimmerman, H. The OSI reference model. **Proceedings IEEE, 71**(12), 1334-1340 (1983).
5. Edwards, J.L. & Mason, J.A. Evaluating the intelligence in dialogue systems. **International Journal of Man-Machine Studies, 27** (1986).
6. Feyerabend, P. **Against Method**. NLB, London 1975).
7. Gaines, B.R. Analogy categories, virtual machines and structured programming. Goos, G. & Hartmanis, J., Eds. **GI—5.Jahrestagung. Lecture Notes in Computer Science, 34**, 691-699. (Springer Verlag, Berlin 1975).
8. Gaines, B.R. On the complexity of causal models. **IEEE Transactions on Systems, Man & Cybernetics, SMC-6**(1), 56-59 (1976).
9. Gaines, B.R. System identification, approximation and complexity. **International Journal of General Systems, 3**, 145-174 (1977).
10. Gaines, B.R. General systems research: quo vadis? Gaines, B.R., Ed. **General Systems 1979**. Vol. 24, pp. 1-9. (Society for General Systems Research, Kentucky 1980).
11. Gaines, B.R. The technology of interaction—dialogue programming rules. **International Journal of Man-Machine Studies, 14**(1), 133-150 (1981).
12. Gaines, B.R. From word processing to image processing in office systems. **Proceedings of International Electrical, Electronics Conference and Exposition, IEEE 83CH1955-4**, 622-625 (1983).
13. Gaines, B.R. Perspectives on fifth generation computing. **Oxford Surveys in Information Technology, 1**, 1-53 (1984).
14. Gaines, B.R. Foundations of knowledge engineering. Bramer, M.A., Ed. **Research and Development in Expert Systems III**. pp.13-24. (Cambridge University Press, 1986).
15. Gaines, B.R. & Shaw, M.L.G. (1984). **The Art of Computer Conversation: A New Medium for Communication**. (Prentice Hall, New Jersey 1984).
16. Gaines, B.R. & Shaw, M.L.G. Hierarchies of distinctions as generators of system theories. Smith, A.W., Ed. **Proceedings of the Society for General Systems Research International Conference**. pp. 559-566. (Intersystems, California 1985).
17. Goguen, J.A. & Meseguer, J. Programming with parametrized abstract objects in OBJ. **Theory and Practice of Programming Technology**. (North-Holland, Amsterdam, 1983).
18. Habermas, J. **The Theory of Communicative Action: Reason and the Rationalization of Society**. (Beacon Press, Boston 1981).
19. Klir, G.J. **Architecture of Systems Problem Solving**, (Plenum Press, New York 1985).
20. Lakoff, G. & Johnson, M. **Metaphors We Live By**. (Chicago University Press 1980).
21. Moto-oka, T., Ed. **Fifth Generation Computer Systems**. (North-Holland, Amsterdam 1982).
22. Popper, K.R. Epistemology without a knowing subject. Van Rootselaar, B., Ed. **Logic, Methodology and Philosophy of Science III**. pp. 333-373. (North-Holland, Amsterdam, 1968).
23. Rosen, R. **Anticipatory Systems**. (Pergamon, Oxford, 1985).
24. Schilpp, P.A., Ed. **The Philosophy of Karl Popper**. (Open Court, Illinois 1974).
25. Schutz, A. & Luckmann, T. **The Structures of the Life-World**. (Heinemann, London 1973).
26. Shaw, M.L.G. & Gaines, B.R. Does the human component in a network have a protocol?. **Proceedings of International Electrical, Electronics Conference and Exposition, IEEE 83CH1955-4**, 546-549 (1983).
27. Taylor, M.M. Layered protocols for computer-human dialogue: I Principles. **International Journal of Man-Machine Studies, 27** (1987).
28. Ulrich, W. **Critical Heuristics of Social Planning**. (Haupt, Bern 1983).
29. Wiener, N. **Cybernetics**. (MIT Press, Cambridge, Massachusetts 1948).