



OFFICE  
AUTOMATION  
CONFERENCE

MOSCONE CENTER  
SAN FRANCISCO  
APRIL 5-7, 1982

afips

---

# 1982 Office Automation Conference Digest

April 5-7, 1982  
San Francisco, California

Conference Chairman: Hans D. Puehse  
Program Chairperson: Ruann E. Pengov  
Digest Editor: Jim Sutton

afips  
PRESS

Sponsored by  
American Federation  
of Information  
Processing Societies

---



## TOWARD HIGH-PERFORMANCE KNOWLEDGE WORKERS

Douglas C. Engelbart  
Tymshare, Inc.,  
Cupertino, California

### INTRODUCTION

Among the on-line knowledge workers of tomorrow, there will be found as always a wide distribution both in personal motivation and flexibility, and in organizational roles and responsibility levels. In this view of the future, two things stand out for me: the workstations and work products of all of the workers must be interconnected; and special roles for high-performance knowledge workers within this inter-linked organizational and informational network will be extremely important. This paper outlines a framework stemming from this perception toward developing high-performance knowledge workers as part of the evolutionary strategy of a knowledge organization.

In the early 60's when I began active, funded research in this area, well before the term "Office Automation" had emerged, I referred to my work as "Augmenting the Human Intellect." (References [1] and [2] summarize events and results for me and my co-workers over the intervening years.)

About ten years ago I re-named our pursuit, after reading Peter Drucker's discussions [3] about "knowledge workers," "knowledge organizations," and "knowledge industries." It seemed that a better term for the work would be "Augmenting the Knowledge Worker." From this new perspective, a natural image emerged of a "Knowledge Workshop" as the place where a knowledge worker does his work and where, if we extended his tools, his means of collaborative communication, his working methods and his organizational roles, we could speak of an "Augmented Knowledge Workshop."

## WORKSHOP ARCHITECTURE

### General Features

It seems inevitable that, as depicted in Figure 1, there will be a combination of local, high-speed networking (Electronic PBX and Local-Area Network) together with higher-level networks (private and public) which will interconnect workstations and the many tools and services within an organization's "whole workshop." The effect will be as though there is a giant communication bus, where some elements seem far away (i.e., a slow or expensive communication path) and some seem very close (i.e., a fast and cheap communication path).

For the purposes of this discussion, let us put aside concerns for how much processing power and storage capacity should be built into the workstation, or where any particular programs or data should reside.

Let us instead consider the following principles, relative to supporting high-performance workers and integrating their capabilities into the larger organization:

- Their workstations should have access to many tools and services, assumedly provided by a number of distributed sources around this network, including both those newly implemented and those that have long existed and will be slow to disappear.
- The collection of tools and services for each worker must be integrated into a coherent whole into his "augmented knowledge workshop."
- Each worker should have access his full complement of tools, services, and personal working files from other workstations

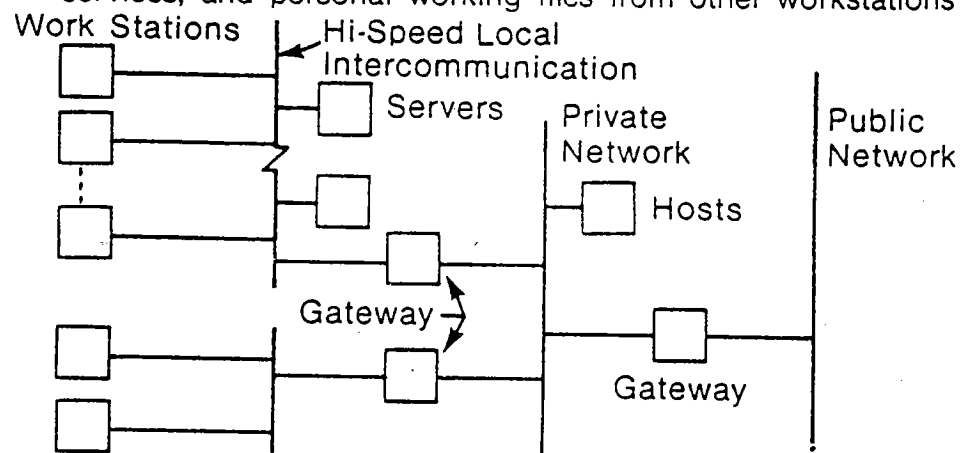


Figure 1. The workstations, computers and data bases for most large organizations will look something like this, and will connect to the outside world via at least one public network.

away from home base, even across the country so he can carry on with his work wherever he happens to be. (It would be a silly rejection of available communication technology to do otherwise.)

- This whole arrangement must provide pragmatically for continuing evolution of command language, tool and service functions, terminal hardware, processor horsepower, application packages and their support computers, etc.

(See Reference [4] for a full development of such principles, and for the foundations for the architecture described below.)

## Basic Organization of the Architecture

The over-all architectural approach that we adopted has four major components, as shown in Figure 2 and summarized below. They are all operational today as part of Tymshare's AUGMENT system.

1. A User Interface System (UIS) to handle the interface between the user's terminal and the interactive programs. (References [5] and [6] provide a detailed description of the implementation and utilization of the UIS.)

The UIS takes care of all command-language dialog and all connection protocols. It also provides a uniform interface between the tool and the terminal to ensure that the user will (as nearly as possible) get the same treatment on a variety of terminals.

It interacts with an individual's user-profile file, to provide interface styles tailored to the needs and preferences of that individual.

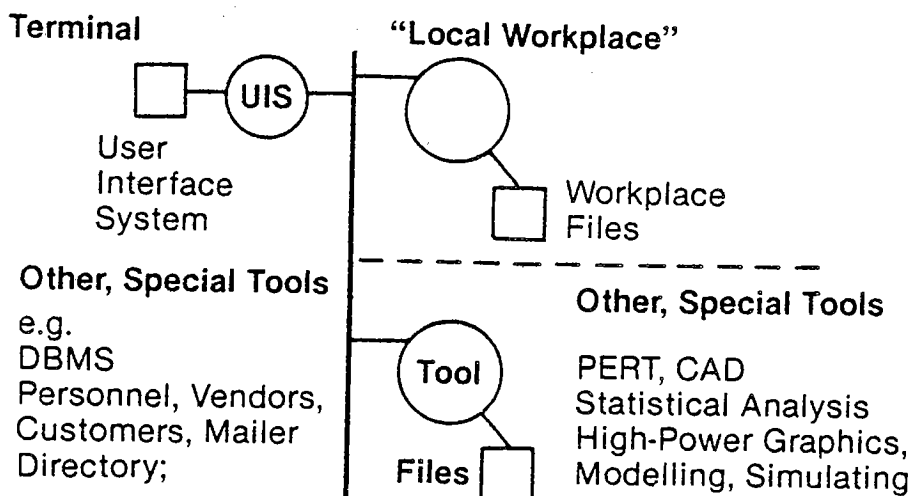


Figure 2. A user at a given terminal will "see" this kind of connection, looking "through" his UIS at his "local workplace" and beyond to the other, special tools that may be located anywhere on a connected network.

It provides a reach-through service to non-AUGMENT systems, and can optionally translate between the command language of a foreign-program modules and a command language designed to meet the user's particular needs. The user's command languages as translated for a number of different "foreign" systems can be designed for mutual consistency, to provide an important coherence in language and function.

It provides an adaptation to different terminal characteristics, allowing users to access their work from different terminals, and enabling application programmers to develop their software as though it were to serve a virtual terminal.

2. A Procedure-Call Protocol (PCP) to provide for effective communication between processes on the network. (Reference [7] gives a thorough, detailed treatment of this "PCP approach".)

This protocol makes possible the implementation in each host of an application-independent, network run-time environment making remote resources accessible at the functional level essentially as though via a procedure call within a one-host application system. It greatly enhances the application programmer's flexibility; makes remote resources usefully accessible to other programs (not just to human users); significantly eases the problems of evolutionary changes within the network; and immensely improves the flexibility with which tools and services can be provided to the user.

3. A Core Workshop the user's own "Local Workplace," a basic collection of tools and services that a knowledge worker generally needs, regardless of his professional specialty.

The user feels that this is his "office," where in a familiar, consistent and effective environment he can do most of his editing, studying, information management, mail management, etc. The AUGMENT Backend was designed to provide these core functions (and in addition has many features which reward a practised user with significant gains in speed and flexibility).

The model in the user's mind is that he does most of his work here, and will "reach through" this "home workshop" to access other tools and services. There is special payoff for effective, flexible capabilities in this core workshop, where the user will spend a large proportion of his on-line time and can steadily acquire more of the available techniques toward higher performance.

4. Other Special Tools with their own file conventions, operating systems, etc.

A rich and ever-growing mix of data bases, application programs and special services will want to be "reachable" in a coherent manner by ever-more of the knowledge workers in a larger organization especially the higher-performance workers.

It is important to support the evolutionary integration of these services into coherent, composite tools systems. AUGMENT's implementation enables application-support programmers easily to provide customized mixes of function and command terminology for special classes of users even for an individual user.

The general case, to be expected and probably encouraged, will find a variety of different hardware elements (terminals, personal computers, minis and large main frames, etc.) and a mix of software (different vintages, vendors, file conventions, terminology, user languages, help conventions, etc.).

## Elements of the User Interface System

In Figure 3 are shown the main software modules (circles, ellipses) and support-file items (rectangles) involved when the User Interface System supports a user's access to a tool that is adapted for direct, "procedure call" service. The AUGMENT Backend is designed this way, and can work with full capability when the UIS and the Backend are separated by a network connection. This is true for any application system that has a procedure-call interface, regardless of the programming language and run-time environment, providing a suitable PCI module is implemented in its host computer to translate between the PCP and the particular procedure-call protocol for that application system.

The main UIS module is the Command Language Interpreter (CLI), interpreting each action by the user and responding with screen-action feedback or calls to the Backend tools for service, according to the particular Command Language in effect.

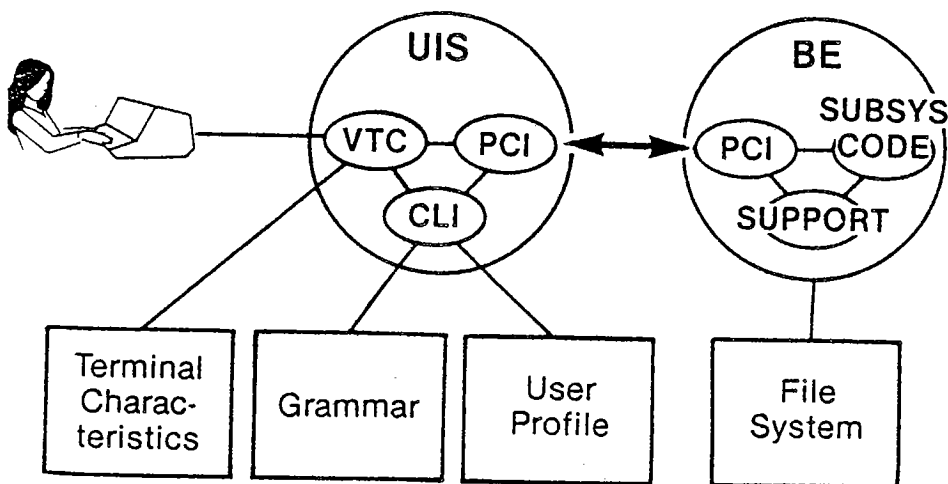


Figure 3. When using the Procedure Call Protocol to interact with a backend tool, the User Interface System (UIS) will employ three special software modules and three special control files.

There are likely to be many UIS-Grammar files lying around, each being a compact, specially coded specification of a particular Command Language. When attached to the CLI, a particular Grammar file determines the command terms and the feedback on the terminal screen, as well as the service-call and data-transfer interaction with the Backend tools.

For any given user, there will be one User Profile file attached to the UIS to specify the particular set of options which that user desires in the action of the CLI e.g., style of command recognition, amount and type of feedback, formatting defaults, initialization status, escape-code assignments to particular keys, etc.

It is an administrative decision whether or not a particular user is provided with commands for changing his profile file.

The Virtual Terminal Controller (VTC) module lets the rest of the UIS operate as though serving a standard, "virtual" terminal, translating back and forth to/from the signals of whatever "actual" terminal is connected.

The characteristics of the particular terminal are packed into the special "Terminal Characteristic" file one such for each different type of terminal that may be interfaced. For most of the modern terminals, this file is selected and installed automatically from interactions between the UIS and the terminal.

The UIS Process Communication Interface (PCI) allows the CLI to interact with the Backend tools making service requests and receiving the results as though it were making sub-routine calls in a "virtual" application-system environment.

In the general case, the UIS PCI would translate the UIS signals back and forth to/from a "universal procedure-call protocol" suitable for network interchange; a particular Backend tool (application system) would employ a version of the PCI that translates in turn back and forth to/from that tool's internally employed procedure-call protocol.

## **Foreign-System Reach-Through**

Figure 4 shows the special provision for reaching through to "foreign" systems that do not provide a procedure-call interface i.e., systems that can only be utilized by character-stream I/O as from a terminal. The Reach-Through Interface is a special module that can be programmed for the specific character-stream interactions of a given tool for eliciting from the tool the equivalent results as expected by each procedure call sent to that tool by the CLI.

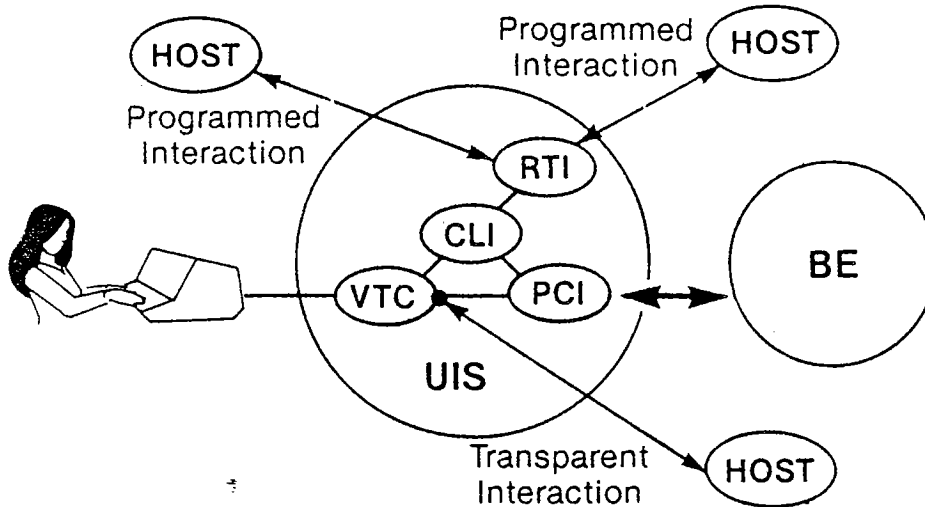


Figure 4. When interacting with a backend tool not equipped for procedure-call interaction, the UIS can employ either programmed interaction via its Reach-Through Interface (RTI), or provide the user with direct, transparent connection.

In such a case, the UIS can interact with the Backend tool as though it (the UIS) were a terminal effectively translating between the CLI and the flow of characters back and forth to/from the tool, to call for service and to receive the results.

Seemingly inefficient, yet this “programmed-interaction” reach-through mode provides for an effective translation between the command language of that foreign tool and the UIS Command Language where the latter may be designed with verbs and nouns etc. to fit the special usage and to be compatible with the rest of the grammar, vocabulary, and conceptual-model characteristics designed to serve this class of users as their coherent knowledge workshop.

This enables the coherent integration of many older systems, many of which will live on for years.

As an alternative mode of interacting with a foreign system through its terminal I/O, the UIS can connect the foreign-system link directly to the Virtual Terminal Controller (VTC) to provide interaction as though the UIS were “transparent.”

## Shared-Screen Conferencing

Figure 5 shows an interconnection mode, between two instances of UIS modules, whereby both terminals can share the screen content of one of them. Each VTC module converts the virtual-terminal screen image to the correct form for its connected terminal, so this shared-screen conferencing will work for dissimilar terminals.



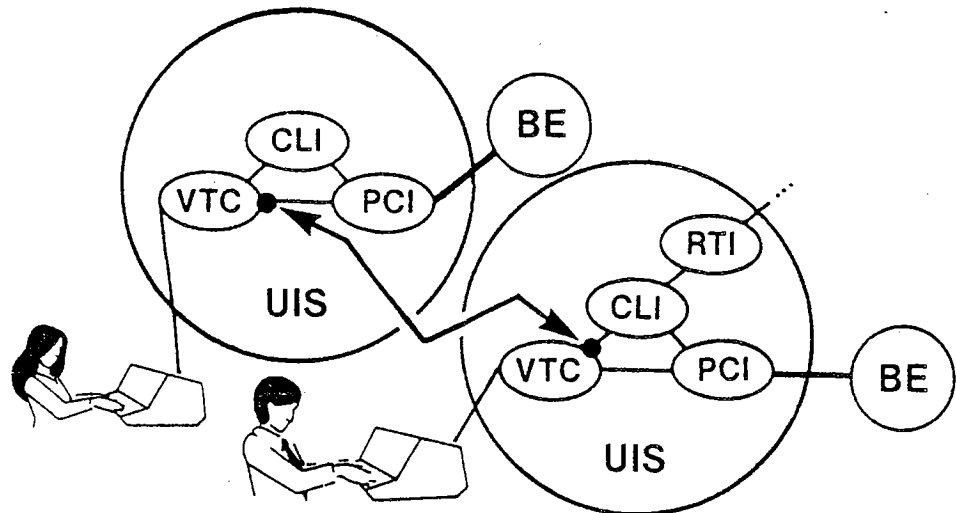


Figure 5. When employing their respective UISs in the shared-screen conferencing connection, two or more users can collaborate closely on whatever job the "showing user" has going.

This mode is established in response to a suitable set of commands by the participants, and in principle any number of users can have such a connection made to their UIS modules so that User A can in real time show the dynamic workings of his screen to them all—no matter what command language and tool system he is using.

At his option, User A can pass control to User B, thereafter what everyone watches are the effects of commands from User B's terminal and VTC acting through User A's CLI upon A's active jobs and files.

In its usual employment, this conferencing mode is used in conjunction with simultaneous telephone dialog. It will work between any two users connected by a network path. (Reference [8] gives a fairly complete description of an earlier form of this "shared-screen teleconferencing.")

## THE OVER-ALL AUGMENTATION SYSTEM

### The Categories of System Elements

Here, from my framework, are the major elements involved in "augmenting" our knowledge workers and their organizations. For this purpose, a "craftsman" metaphor seems directly applicable—considering that our knowledge workers must be very much the professional craftsmen.

- A. Tools: Craftsmen benefit from balanced collections of well-designed tools
- B. Methods: To be effective, tools must be used with well-polished work methods

- C. Skills: It takes practised skill to exercise a competent blend of tool and method
- D. Knowledge: True craftsmen depend upon much integrated "shop" knowledge
- E. Language OF THE CRAFT: Craftsmen need an effective language to discuss, teach, plan and collaborate among themselves (i.e., to do their "shop talk").
- F. Training: To develop an effective group of craftsmen in a planned way requires explicit training, in all of the above elements
- G. Organization: Role differentiation and organizational structure are necessary for integrating craftsmen effectively into an organization.

### **Tool System and Human System**

For discussion sake, call Category A the "Tool System" and the aggregate of Categories B through G the "Human System." We can immediately note that new technology, no matter how dramatic, contributes directly only to the Tool System.

Over the centuries there has been an immense amount of invention involved in the cultural evolution that brought the Human System to its present state. But its evolution took place with what will have to be described as a very primitive Tool System.

To take advantage of the absolutely radical, emerging Tool-System inventions, it is inevitable that evolution of the Human-System will begin to accelerate. In my view, this is strongly to be encouraged, since the power derived from the Tool System can only come from the way it is harnessed to human endeavors via the Human System.

### **Co-Evolution**

The optimum design for either the Tool System or the Human System is dependent upon the match it must make with the other. There is a high degree of mutual dependence. But it seems that the Tool System is or soon will be "out of control" in the sense of our being able to design its target state, say for five years hence. And we possibly never will know how to "design" this Human System. So to be pragmatic about it, we can at best work in a "guided-evolution" mode for each of the sub-systems.

So, the ultimate capability of the larger "Augmentation System," and therefore the performance level of the knowledge workers and knowledge organizations of the future, will improve only through the co-evolution of these two sub-systems. A disastrous default mode would be for the perceptions of the technologists and the market-oriented product planners to steer the evolution of the Tool System, and leave the Human System to adapt in its trail. There is no

practical worry that the evolution of the Human System will drive that of the Tool System; it is inconceivable that the Human System could be served by analysts, inventors and entrepreneurs with the same fierce intensity as for the Tool System.

The practical worry is that there won't be enough perception of payoff from investing in explicit, conscious invention and evolution in the Human System, and that we will drift toward the above default mode.

It is something of a bind—our culture hasn't really developed an acceptance for cultural progress to anywhere near the extent it has for progress in the technological and material sense—and without a solid perception and acceptance that conscious evolution of such as this Human System (primarily a cultural matter) will pay off, we are not likely to become particularly effective at it. So it would seem that we need to invest an extra degree of attention and resource toward developing the perception that this Human System is not only acceptable but has a very high payoff. THEN we probably could get moving toward a balanced co-evolution.

## **SO, WHY TALK ABOUT HIGH-PERFORMANCE KNOWLEDGE WORKERS**

There is a first-order answer to this question. It makes sense, at least from my viewpoint, to aim for a balanced distribution among the knowledge workers in an organization, in terms of the level of knowledge-work performance targeted for different roles. In this view then, a certain proportion of research, development and implementation investment should be made toward making really significant improvements. This would involve special attention for such roles, over both the Tool System and the Human System.

And there is also a very important, second-order answer. The most effective strategy that I can think of, toward developing the perception and acceptance of "progress" in the Human System, is to invest in pursuit of truly high-performance for selected knowledge-work roles. The best roles for this purpose would be those that would expose important stakeholders to the EXPERIENCE of truly high performance, by BEING THERE when that high performance is being exercised on activities relevant to their workaday world.

As a general strategy then, we would aim for specially equipped and trained teams to be connected into the workshop networks of large organizations, to perform roles that lend themselves best to early pursuit of especially high performance, and where there would be an appropriate visibility, identification, and sense of relevance for the organization's trend setters.

## CONCLUSIONS

We can reasonably hypothesize that a startling degree of improvement may be obtained in the performance level of knowledge organizations and their individual knowledge workers. And further, that in order to obtain this we must attend to changes in both the Tool System and the Human System.

If this hypothesis were to be proven valid, it would be of immense importance for a problem-laden society to have acted on it. It doesn't seem that we would have to risk much to test it out over the next decade. A very small proportion of what is being invested in the "easy to learn" level of Office Automation, if explicitly directed toward pursuing high augmented-human performance, would have a notable effect.

Architectural features such as described above seem necessary anyway to support the natural evolution of Office Automation, even without any special emphasis upon high-performance workers. A salient point is that these features also can support the accelerated evolution of individuals and groups, who can still work effectively with the rest of the organization, but where through their own efforts or through planned investment by the larger organization they have extended more rapidly than the rest the development of their augmentation categories—tools, methods, skills, etc.

And what is also important about these features is that they provide for the harmonious co-existence, within the same organizational environment, of knowledge workers of all levels of performance. The high-performance organization of the next decade must make do with many degrees of aspiration, talent and training, and must accommodate a wide spectrum in its workers' performance levels.

And it is also important to note that architectural characteristics of the organization's knowledge workshop will have a notable effect upon the co-evolution rate of that can be achieved.

## ACKNOWLEDGEMENTS

The concepts and the system described above have evolved over more than two decades, greatly aided by the research sponsorship of a number of organizations. Until 1978, at SRI International, research sponsorship by The Air Force Office of Scientific Research provided three years of critical conceptualization and planning support, from '59 through '62; DARPA's Information Processing Techniques Office, NASA, and the Air Force RADC contributed significantly until 1978, when SRI sold its rights to the system to Tymshare, Inc. There, while bringing it into the commercial market, the company has supported further conceptual and development work. During this more than two decades, probably a hundred different people have contributed directly, very significantly affecting the architecture and its implementation, and probably even affecting the way I see these things.

## REFERENCES

1. Engelbart, D. C., "Toward Integrated, Evolutionary Office Automation Systems", Proceedings of the 26th Joint Engineering Management Conference, Denver, CO, 1978, pp. 63-68.
2. Engelbart, D. C., "Evolving the Organization of the Future: A Point of View," published in the book, "Emerging Office Systems Issues," ed. Robert Landau, Ablex, 1982 (Proceedings of the Stanford International Conference on Office Automation, March 1980).
3. Drucker, Peter F., "The Age of Discontinuity: Guidelines to Our Changing Society," Harper and Row, New York, 1968.
4. Engelbart, D. C., Watson, R. W., and Norton, J. C. "The Augmented Knowledge Workshop," AFIPS Conference Proceedings, Volume 42, pp. 9-21, National Computer Conference, June 4-8, 1973.
5. Irby, C. H., "The Command Meta Language System," AFIPS Conference Proceedings, NCC Vol. 45, 1976.
6. Watson, R. W., "User Interface Design Issues for a Large Interactive System," AFIPS Conference Proceedings, NCC Vol 45, 1976, pp. 357-364.
7. White, J. E., "A High-Level Framework for Network-based Resource Sharing," AFIPS Conference Proceedings, 1976, NCC, Vol. 45.
8. Engelbart, D. C., "NLS Teleconferencing Features: The Journal, and Shared-Screen Telephoning," Proceedings of 1975 COMPCON, Washington D.C., September 1975.