

MIT/LCS/TM-195

RECORD OF THE WORKSHOP ON RESEARCH  
IN OFFICE SEMANTICS

Gerald R. Barber

February 1981



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LABORATORY FOR COMPUTER SCIENCE

February, 1981

Record of the Workshop on  
Research in Office Semantics

Gerald R. Barber

Abstract: This paper is a compendium of the ideas and issues presented at the Chatham Bars Workshop on Office Semantics. The intent of the workshop was to examine the state of the art in office systems and to elucidate the issues system designers were concerned with in developing next generation office systems. The workshop involved a cross-section of people from government, industry and academia. Presentations in the form of talks and video tapes were made of prototypical systems.

Key Words: Office Automation, Knowledge-Based Office Systems

Acknowledgements: This report describes research done at the Laboratory for Computer Science of the Massachusetts Institute of Technology. Support for this research was provided by IBM.



# Record of the Workshop on Research in Office Semantics

## Table of Contents

|   |    |
|---|----|
| 1. Introduction   | 0  |
| 2. Research in Workstation Network Semantics  | 1  |
| 3. Implementation Strategies for Office Automation Systems                                  | 3  |
| 4. Office Streamlining  | 4  |
| 5. The System for Business Automation (SBA)   | 4  |
| 6. On Supporting the Use of Procedures in Office Work                                       | 5  |
| 7. Odyssey: A Knowledge-Based Assistant   | 6  |
| 8. Pie: A Network-Based Personal Information Environment                                    | 6  |
| 9. Simple Semantics for Naive Users   | 7  |
| 10. Voice and Text as Alternate/Competing Media for Communication in the Office Environment | 8  |
| 11. The Integrating Role of the Workstation   | 9  |
| 12. Towards the Integrated Interactive System   | 10 |
| 13. Linguistic Considerations for the Design of Integrated Electronic Office Systems        | 11 |
| 14. Office Work as Practical Action   | 12 |
| 15. Additional Discussions  | 13 |
| I. The Participants   | 13 |



# Record of the Workshop on Research in Office Semantics

June 15-18, 1980

Chatham Bars Inn  
Chatham, Cape Cod  
Massachusetts

*Edited by Gerald R. Barber*

## 1. Introduction

This paper is a compendium of the ideas and issues presented at the Chatham Bars Workshop on Office Semantics. The intent of the workshop was to examine the state of the art in office systems and to elucidate the issues system designers were concerned with in developing next generation office systems. The workshop program committee consisted of Michael Hammer, Carl Hewitt, Howard Morgan, and Jeff Rulifson. The workshop was sponsored by the MIT Laboratory for Computer Science.

The workshop involved a cross-section of people from government, industry and academia. Presentations in the form of talks and video tapes were made of prototypical systems. The participants discussed the systems they were currently developing. The discussions concentrating on the breadth of application to the office, the problems involved in developing the application and the necessary hardware and software technology.

The primary intent of this paper is to present a collection of the ideas and issues discussed. It is hoped that this will be useful as an indicator of some of the work in the field of office systems. The paper is composed of a series of abstracts for most of the talks given. In some cases references for further reading are included. Brief descriptions of some discussions for which abstracts were not available at the time of printing are also given. The appendix contains a list of the participants and their respective organizations.

As a guide to the abstracts in the rest of this paper shown below are various subjects under which the papers can be grouped.

- **Understanding Office Work.** This subject is concerned with understanding what is the work done in offices and how this work may be supported with modern technologies. The abstracts addressing this topic are: *Research in Workstation Network Semantics*, *On Supporting the Use of Procedures in Office Work*, *Office Work as Practical Action*.
- **Technologies for the Office.** New tools for use in the office. The papers addressing this subject are: *The System for Business Automation (SBA)*, *Odyssey: A Knowledge-Based Assistant*, *PIE: A Network-Based Personal Information Environment*, *Voice and*

*Text as Alternate/Competing Media for Communication in the Office Environment.*

- **Implementation Strategies.** The proper ways to implement an office automation system, taking into consideration the people involved, the organizational structure and the application task. The paper concerned with this subject is *Implementation Strategies of Office Automation Systems*.
- **Design Methodology.** Issues in system design in light of the diversity of interactive system. The paper concerned with this subject is *Towards the Integrated Interactive System*.
- **The Integrated System.** The integrating of functions in the office with an electronic system. *Towards the Integrated Interactive System*, and *The integrating Role of the Workstation* address this issue.
- **The Interface with the User.** The user interface and how the user views the system she/he uses. This is discussed in *Simple Semantics for Naive Users* and *Linguistic Considerations for the Design of Integrated Electronic Office Systems*.
- **Analysis.** Analysis of information flow in office system. Techniques for improving the information flow. This is discussed in *Office Streamlining*.

## 2. Research in Workstation Network Semantics

*Gerald Barber and Carl Hewitt*

Workstation Network Semantics is the search for, and development of basic approaches that support negotiation and problem solving activities on a network of workstations. The research has two major thrusts -- *knowledge engineering* and *workstation technology*.

The *knowledge engineering* thrust is to develop: (1) a workstation explanation knowledge base; and (2) an application knowledge base concerning the DOD officer transfer process. Knowledge engineering focuses on basic approaches that support problem solving activity on a network of workstations taking into consideration the goals and constraints derived from the users' organization, the application domain, and the task at hand.

The *workstation technology* thrust is to integrate new technology in high resolution color displays and pointing devices for the explicit purpose of knowledge based support in the context of problem solving and inter-workstation negotiation.

As a step toward understanding the impact of the expanded use of computers in the office we propose to investigate *Workstation Network Semantics*. In our view Workstation Network Semantics encompasses the study of the two dominant structures in the office--the application structure and the organizational structure--and how these structures interact within themselves and between each other. The basis of inter- and intra-structure interaction is through communication of information. Thus communications have content or meaning in terms of the application structure and the organizational structure. We couch this meaning or the semantics of the communications in terms of the effect of these communications on the subsequent behavior of the office system structures.

The application structure concerns the subject domain of the office. It includes the rules and



objects that compose the intrinsic functions of a particular office system. In an office concerned with loans the application structure includes such entities as loans, credit ratings and such rules as criteria for accepting or rejecting loans. The application structure of an insurance company is concerned with insurance policies, claims and actuarial tables. The application structure explains the scope of the functionality an office system has on a subject domain as well as providing a model by which those functions are characterized. The application structure is, overtly, the primary reason for the existence of the office system.

In contrast to the application structure we have the social structure of the office system as an organization. Our concern with this aspect of an office system stems from the fact that the activity in the application domain of an office system is realized by people cooperating in a social system. The structure of this social system involves such aspects of an organization as the roles of the individual participants, the interaction of roles, and the various subsystems that make up the organization. We view the office system as a functioning organism in an environment from which it extracts resources of various kinds and to which it delivers the products of its mechanisms.

To develop a formalism within which to describe the structures in an office system we draw on ideas and theories from the field of Artificial Intelligence. The formalism we are developing allows us to embed knowledge within a computational system and reason using this knowledge. This allows us to describe and reason about the application structure and the organizational structure.

Problem solving is a pervasive aspect of workstation procedures which has not received adequate attention until very recently. Understanding this problem solving activity is a prerequisite to developing systems which perform tasks in a distributed environment, that previously were not amenable to computer processing. Several situations give rise to problem solving activity: problem solving is often required within the application domain. Decisions are made concerning the best way, according to some criteria, of obtaining some result. A common task requiring problem solving is to diagnose abnormal results of a workstation procedure. In this case it is necessary to reason about the progress of a procedure in an effort to pinpoint the cause of the anomalous behavior. Once this is done, further reasoning is necessary to determine what all the abnormal effects of the procedure are and how to compensate for them.

The knowledge-based workstation network helps support problem solving activity. Descriptions of past activity help diagnose abnormal workstation procedures and descriptions of consequences of proposed activities aid in the problem solving process. A knowledgeable workstation tracks goal and constraint information for the task at hand.

#### References:

Attardi G., G. Barber, and M. Simi  
Towards an Integrated Office Work Station  
in *Automazione e Strumentazione*  
No. 3, March, 1980.

Barber G. R.  
*Reasoning About Change In Knowledgeable Office Systems*  
MIT AI Working Paper No. 200, August, 1980.

Hewitt C., G. Attardi, and M. Simi  
 Knowledge Embedding with a Description System  
 in *Proceedings of the First Annual AAAI Conference*  
 December, 1979.

### 3. Implementation Strategies for Office Automation Systems

*Jack R. Buchanan*

Taken from a paper appearing in:  
*The Proceedings of the NCC Office Automation Conference,*  
 Atlanta, Georgia, March 3-5, 1980.

Wide spread proliferation of advanced office automation systems will not happen by itself nor as a spontaneous result of the rising generation's increased educational level any more than automobiles would have been successfully introduced in this country without dedicated technologists and businessmen making them safe, reliable, easy to use, economic and functionally what the public wanted. A critical aspect of this proliferation is the development of implementation strategies. Strategies for a particular organization will depend on the technology being used externally as it relates to the technology mastered internally, the application task, the organizational structure and the people involved.

The process of developing implementation strategies for office information systems begins during the planning phase and continues throughout the system life cycle until relatively stable operational status is achieved. Implementation may be said to have occurred when the *right* system has been *properly designed and used* by an organization to displace costs or labor, or to provide significant new functionality. The definitions of when particular systems are "implemented" may differ according to system or organization, but for each project this definition should be explicitly formulated to produce a "defined setting." Some research in particular application areas aimed at studying and formulating implementation strategies has been carried out.

Formal implementation will be facilitated following some of the strategies here discussed along with traditional approaches to user training, phased parallel operation, data conversion and modifications of plans, designs and code as the systems converge to the office requirements. The challenges in automating office administrative tasks appear to be that of integrating the basic generic functions to match these tasks and conversely articulating the higher level tasks as combinations of operations currently feasible. If this is accomplished, an operational understanding of the organization is gained which is fundamental to successful implementation.

#### References:

Buchanan J. R. and R. D. Fennel,  
 Model-Directed Information Systems for Management of the Federal Courts,  
*Harvard Business School Working Paper # 79-80.*

Buchanan J. R. and R. G. Linowes,  
 A Framework for Understanding Distributed Data Processing,  
*Harvard Business School Working Paper # 79-78.*

Buchanan J. R. and S. D. Neeleman,  
 The Design of a Legal Document Construction System,  
 in *Proceedings of the MIDCON Computer Conference*,  
 Dallas, Texas, December, 1978.

## 4. Office Streamlining

*Clarence Ellis, Robert Gibbons and Peter Morris*

Taken from the Introduction to:  
 Institut de Recherche d'Informatique et d'Automatique,  
*Proceedings of the International Workshop of Integrated Offices*  
 Versailles, France, November 1979.

This paper describes a mathematical technique for streamlining the information flow within offices. Although the technique performs a number of useful transformations and optimizations on an office flow diagram, its primary purpose is to derive a normal form description of any office and then to use this to derive other forms of the office which are minimal in certain well-defined ways. These minimal forms of the office description show the basic necessary information flows and the invariant information requirements that must be met in any realization of a specific set of office functions.

The work described is part of a joint effort by the Office Research Group and the Analysis Research Group at the Xerox Palo Alto Research Center to describe and analyze information flow within offices. The technique is supported by a rigorous mathematical model called Information Control Nets. We describe an example, the loan office, and show how to convert this office into its normal form and its minimal form. We then summarize these results and show other possible benefits of expressing an office in normal form terms.

### References:

Ellis C. A.  
 Information control Nets  
 in *Proceedings of the ACM Conference on Simulation, Measurement, and Modeling*  
 Boulder, Colorado, August 1979.

Cook C. L.  
 Streamlining Office Procedures  
 in *Proceedings of the 1980 National Computer Conference*  
 Anaheim, CA. May 1980.

## 5. The System for Business Automation (SBA)

*S. Peter de Jong*

The System for Business Automation (SBA) is a system which allows the end-user as well as programmer to directly describe their office and business applications on the computer. It contains a unified language which replaces the various languages and functions currently needed to implement an application: Command language, programming language, data base, screen manager, and

communications. The approach is for the end-user to deal with the two-dimensional objects he is familiar with when he performs the application manually: tables, forms, charts. In addition to supporting an individual end-user's application, the SBA system mirrors the behavior of a business organization consisting of many people, performing interrelated applications in multiple internal and external locations. The distributed applications communicate by electronic mail - consisting of tables, forms, and programs - which flow around the network, triggering the applications in various business locations (nodes). The various nodes represent people performing business roles, and organizational entities such as departments. All the applications and communications will be specified by the same end-user who would perform the application manually.

The SBA system is constructed out of abstract objects called BOXes. Each box is independent and can execute concurrently. Each box contains the operations necessary to support the object and object's data. Each object or collection of related objects forms a complete application system. The SBA system grows in power with the addition of independent objects. There is a common SBA language for constructing and executing objects, although each class of objects can have its own end user interface. Most of the focus of the project has been on the following relational languages: Query-by-Example, which is a sub-component of SBA, and the SBA programming language. Other classes of objects include semantic objects which use semantic information to raise the level of the user interface and analogue objects whose operations consists of moving lines on the screen (e.g. Gantt charts, peg boards). An application area can have its own language, manufacturing objects have operations as explosion and pegging; accounting objects have operations as posting and aging.

#### References:

de Jong, S. P.

The System for Business Automation (SBA): A Unified Application Development System  
in *IFIP Congress 80 Proceedings*, October, 1980.

*IBM Query-by-example Terminal User's Guide*  
Systems Reference Manual SH20-2078  
IBM Data Processing Division, White Plains, NY.

Zloof M. M., S. P. de Jong

The System for Business Automation (SBA): Programming Language  
in *Communications of the ACM*, Vol. 20, No. 6, June 1977.

## 6. On Supporting the Use of Procedures in Office Work

*Richard E. Fikes and D. Austin Henderson, Jr.*

In this paper, we discuss the problems involved in developing computer-based systems that support the specification and use of procedures in office work. We begin by arguing that the real work of carrying out office procedures is different in kind from the standard computer science notions of procedure "execution". Specifically, office work often requires planning and problem solving in particular situations to determine what is to be done. This planning is based on the goals of the tasks with which the procedures are associated and takes place in the context of an inherently open-ended body of world knowledge. We explore some of the ways in which a system can provide support for such work and discuss the requirements that the nature of the work places on such support systems.

We argue that paradigms and techniques developed in the Artificial Intelligence research fields of planning and knowledge representation are useful for meeting those requirements, and that the requirements, in turn, present new research problems in those fields. Finally, we advocate an approach to designing such office systems that emphasizes a symbiotic relationship between system and office worker.

## 7. Odyssey: A Knowledge-Based Assistant

*Richard E. Fikes*

This paper describes an investigation into the problems of representing and using task domain knowledge in office information systems to assist with the filling out of a collection of electronic forms. In particular, a demonstration system called Odyssey is described which aids in the preparations for a business trip. The system uses knowledge about trip planning to maintaining consistency of the information in the data base, infer additional records and values for the data base, reformat field entries on forms, correct spelling errors, etc. We discuss the "frame oriented" style of programming used to design and implement Odyssey that combines "frame-structured" knowledge representation and "object oriented" programming. We focus on the problems involved with allowing the user at any time to enter or change information in any of the forms. A dependency maintenance facility is described that deals with those problems by allowing the application of domain knowledge to data whenever it enters the data base and the removal of derived results whenever the data used in the derivation is removed or changed.

## 8. Pie: A Network-Based Personal Information Environment

*Ira Goldstein*

A personal information environment (PIE) has been implemented collaboratively with Dan Bobrow that supports the usual OIS tasks of document preparation, electronic mail, and database management. PIE is unusual, however, in that it supports all of these functions by means of a uniform representation of office information. This representation is an extended semantic network. This network is employed to represent specific facts (personnel data, appointments), generic information about different kinds of entities (constraints, defaults), and procedural knowledge regarding the functions associated with different entities (summarizing personnel data, scheduling appointments). The uniform application of this network structure has proved to be both powerful and comprehensible: powerful because so many OIS tasks can be represented in this fashion, and comprehensible because the user need only understand a few concepts related to the creation and manipulation of networks. Data and programs can be treated in a similar fashion. The result is that PIE is one of the few office information systems that bridges the gap between user and programmer.

### References:

- Bobrow, D.G. and I. P. Goldstein  
 Representing Design Alternatives  
 in *Proceedings of the Conference on Artificial Intelligence and the Simulation of Behavior*  
 Amsterdam, July, 1980.

Goldstein, I.P. and D. G. Bobrow, D.G.  
 Extending object-oriented programming in Smalltalk  
 in *Proceedings of the Lisp Conference*  
 Stanford, August, 1980.

## 9. Simple Semantics for Naive Users

*Reinhard Kofer*

Offices are operated by and provide services for a multitude of people, some of whom are more expert and some of whom are more naive. Expert users are sufficiently well taken care of by the current state of the art in programming languages and program development systems. Things are markedly different when "naive: (or inexperienced resp. not currently trained resp. intellectually limited resp. "non-expert" in any other conceivable way) personnel is supposed to interact directly with a computer based office.

Different philosophies of day to day practical action are possible:

1. "Guide" the user (resp. coax, prompt, force him to do things no matter whether he likes it or not). This is a valid approach for grossly inexperienced users. "Unfortunately" most of them don't stay in this condition, unless they are downright "stupid".
2. "Natural Design" of the computerized work environment (resp. task oriented tools and data objects resp. no operational overhead as compared to average effort for manual task execution). This is a valid approach for "static" offices with well defined flow of control and information and create stability over time. "Unfortunately" you need scrupulous analysis before you can set up reliably operating systems.
3. "User driven modeling" of the computerized work environment (resp. alterable data objects and external representations resp. adaptable control and flow of data). This is a valid approach for users who "have outgrown their children wear". "Unfortunately" people that are alert and judicious derive more profit from this approach than "uncreative plodders", otherwise these people have been the initiators of change anyhow - experts or not...
4. "Assistance by intelligent machines" i.e. casewise predictions of probable user-demands, given previous hints and experience on the machine's side. "Unfortunately", this is utopian in general circumstances, regardless of the possibility of limited inferred help via small real world models.

Of course there is no sharp boundary between these "methods" of user-friendly system interaction, but extreme representatives of 1 resp. 4 can serve to demonstrate the enormous spectrum of opinions in the new field of computerbased office automation.

The author is a firm adherent of method 3, but tries to incorporate necessary properties of the other methods in his "user-friendly solution" nevertheless. The contribution describes a semigraphic interaction method that serves as an advanced job control and command language (henceforth called JCCL) and its interplay with an advanced specification tool that contains an executable nonprocedural specification language (henceforth called NPSL).

Of the many problems that arise around such an approach the following two are addressed in particular:

1. What is the semantics of the NPSL?
2. Is it adequate for the "naive" user?

In several short examples it is demonstrated, that the NPSL can be executed in terms of the JCCL. Consequently the user can transfer his operative model as a descriptive model.

In an "animated" sequence of pictures it is demonstrated that JCCL deals with setting up and influencing a task-specific cooperation of so-called "entities" like forms, files, processes and datalinks. These are immediate electronic replicas of well known paper objects - hence "natural".

Special emphasis is put on the "programming by example" nature of the method of interaction. (on that, further publication is planned later.)

## 10. Voice and Text as Alternate/Competing Media for Communication in the Office Environment

*John O. Limb*

Traditionally, voice and text have had specific and non-overlapping areas of application in the office. Voice was used for simultaneous communication; text was used for non-simultaneous communication. With the advent of computer mail with its greatly reduced response time relative to the postal service, there is the potential for non-simultaneous text to replace some simultaneous voice, particularly where the communication requires no answer or a request is made for specific information. The situation is becoming more confused with the advent of stored-voice services which also permit non-simultaneous communication.

Advantages of text are that it can be scanned by a human who is interested only in specific topics or searched by a computer for keywords. It can be filed easily and retrieved by a header or keyword. Storage requirements are very modest. Voice input, on the other hand, provides a more natural interface and permits a greater sense of communication; the cost of storage and processing are decreasing; scanning of a voice message can be facilitated by speeding up the playback and the ability to jump to marked positions (e.g. "chapter headings"). Schemes for flexibly editing voice sequences similarly to the way one edits text are now being developed.

We may see develop essentially new modes of communication by the synthesis of the two media. For example, text outlines may supplement long voice messages and voice annotation may be combined with long text messages. The use of two media also provides greater flexibility in reaching a user. He or she may much prefer to receive a voice message while in the middle of reviewing a text document than to be distracted by a further text message on the screen.

The whole question of when and how to use voice or text for communication is an area ripe for considerable research. We hope to learn better how to exploit these media both separately and in combination for more efficiently controlling the office environment.

## 11. The Integrating Role of the Workstation

*Howard Lee Morgan<sup>1</sup>*

Much attention has been focused on workstation design within the office automation and decision support systems community. This has mainly been directed at providing better user interface standards, and permitting a user to more easily access different sources of information. Emphasis has been placed on providing the following tools within the workstation:

1. text entry and editing systems
2. electronic mailbox capability
3. electronic filing
4. simple data processing functions
5. network access to other machines
6. administrative support tools
7. myriad of other functions depending on the whim of the research group.

One might point to another role which the workstation can play - in a manner similar to the physical desk and walls of an office. This is what can be termed "the integrating role". This focuses more on the ways in which information from different sources can be brought together for use by the decision maker or worker, than on the straightforward simulation of office tasks (which are a necessary prerequisite).

Comparison of information is one common use of a desk (where two or more papers are laid side by side) which can be aided with a powerful workstation. The window concept embodied in various systems (Wharton, Xerox PARC, Couloris) directly addresses this problem.

Filtering the information that appears on the desk is another role that an intelligent workstation should play. The proper application of database alerting techniques would allow different sources to be monitoring and filtering text and other sources for appearance on an individual workstation.

No single standard computer system is likely to emerge in the next few years. Another important role which a workstation can play is to ease the interface with different information sources. In the desk mode, it is simple to get any piece of paper which is transmitted (via US mail, for example) onto the surface. Workstations have considerably more difficulty in providing the same facility. We must have better methods for designing interface protocols to new systems, ones which permit users to quickly access new information sources.

One possible method for doing this is via "Interface through example", a research project just starting at the University of Pennsylvania. system, users are shown a rich set of examples, and are able to modify A brief description of this proposal is as follows:

Most of us actually learn how to perform a specific task on a computer through the clinical method.

---

<sup>1</sup>Research supported in part by the Office of Naval Research under Contract N00014-75-C-0462.



That is, we watch someone go through the task, or look at an example in a reference or other manual, and then slightly deform the example to fit our particular need. This appears to achieve quicker results than the alternative of using the reference manuals to determine the proper syntax and semantics for the task at hand. Many successful manufacturers provide manuals consisting mainly of examples to assist naive users.

This proposal is, in essence, to automate the above process. Thus, users would be able to examine the examples, and "deform" them slightly to fit their own particular situation. It is hoped that this would lead to more rapid and effective use of their systems.

There are three major problems to be solved in creating a user interface through examples. These are:

1. Choosing a rich enough set of examples
2. Providing a means for the user to find a sufficiently "close" example, and
3. Building a good editing and interface facility for transforming the example into the user query.

It is hoped that a version of the above interface through example system based on color hardware, and a small microcomputer running Unix will be demonstrable early next Spring.

The workstation concept has arisen out of both the task replacement and task enhancement types of activities described by Zisman. One would hope that a clearer descriptive and prescriptive methodology will be available in the future to aid in workstation design. Perhaps a "user semantics" rather different than the "office semantics" (such as Zisman, Tsichritzis, Ellis and Nutt), is required. This "user semantics" must model NOT so much the information flows, but the information perceptions which a decision maker is looking for. The AI methodologies (Actors, frames, etc.) and database alerting methods are applicable here, but other more human factors oriented information (e.g., role of colors), will have to be taken into account.

## 12. Towards the Integrated Interactive System

*William Newman*

One of the most remarkable aspects of interactive computer systems, besides their rate of market growth, is their diversity. Indeed it is difficult to keep count of the uses to which interactive systems are applied: diagnosing illnesses, playing chess, designing clothes, dispensing cash, and training airline pilots are just a few examples. Furthermore, there appear to be no limits to diversity in the user interfaces of these systems: they use keyboards, function keys, touch-sensitive screens, graphical selection, command menus, symbol and voice recognizers; systems just around the corner will offer us eye tracking, holographic displays and quadraphonic audio feedback. All of this diversity helps to maintain the excitement and challenge of designing interactive systems.

This increase in the diversity of interactive systems does not, however, make the task of designing them any easier. Each new system is likely to address a new community of users, whose requirements may not be fully understood. Faced with a poorly defined set of requirements, and a growing plethora of interactive technology, the designer can be forgiven for not knowing exactly how to go about his business. What he needs is a methodology for designing interactive systems; a recent

IFIP-sponsored workshop has been instrumental in drawing attention to the lack of such methodologies, and the difficulty of developing them in the face of continuing diversity in applications and techniques.

The need for a design methodology is indeed severe, and the problem of developing a methodology that can be aimed successfully at such a rapidly moving target is a challenging one. Before attacking the problem, we should consider whether the target is likely to slow down: are applications likely to cease proliferating, and is the technology of user interfaces likely to stabilize? If so, the magnitude of the problem will be dramatically reduced.

The thesis of this paper is that diversity amongst interactive systems is indeed on the wane. This is not due to any reduction in the demand for interactive systems; on the contrary, the increased proliferation of these systems is creating a demand for commonality. The paper discusses *integrated* interactive systems, and addresses some of the relevant design issues.

### 13. Linguistic Considerations for the Design of Integrated Electronic Office Systems

*Sabine Rohlf's*

Office workers without EDP background knowledge often have problems with interactive computer languages. There are two basic reasons for these problems:

1. The grammatical rules of the interactive language are different from the user's natural, i.e. native language and therefore difficult to learn and to memorize. In addition, the rules of the interactive language are not consistent with the rules of natural languages.
2. The semantic and pragmatic concepts associated with terms (e.g. "record") of the interactive language are not understandable for the user, since they are EDP-based and the user cannot relate them to his experience, or because the user assumes a meaning from his experience that is not intended by the designer.

Our research is aimed at improving this situation by developing methods to facilitate communication between users and designers about system concepts and the syntax and semantics of interactive languages. This will also enhance participative design techniques and introduction and use of integrated electronic office systems.

Two methods stemming from linguistics will be outlined to design consistent interactive, command oriented languages, following the principals of natural languages, and to determine a vocabulary for the user language which relates to the user's experience, and which is precise with respect to computer system requirements.

In order to achieve an interactive language that follows grammatical principles of natural languages the structure of the interactive language is described in terms of dependency grammar. For each command, such as the verb "read", the possible dependent objects, such as "file A", and supplements, such as "from line 22 to 24", are specified and analyzed to determine whether they are compatible with possible dependent objects and supplements of natural language use. According to such a procedure, command sequences are avoided that appear grammatically unacceptable to the user and are therefore hard to learn and to memorize.

In order to achieve a vocabulary that is understandable for the user and relates to his experience, the elements of the interactive language are described in terms of generative semantics, i.e. the underlying concepts and the use of the element such as a verb is described, compared to the concepts and use of the element in natural language use, and discussed with the user in order to gain mutual understanding between user and designer. The designer gets a better understanding of the user's needs in terms of language requirements, and the user gets a better understanding of the system's concepts. The overall efficiency and user-friendliness of the system is improved.

## 14. Office Work as Practical Action

*Lucy Suchman*

The first premise of office research is that structures of office work are a more or less direct outcome of procedural specifications. This research premise has a long-standing precedence in management theory, wherein procedural specifications promise a means by which plans can be systematically turned into orderly practice. More recently, computer scientists working on office systems design have found in this understanding of structure in terms of procedures a "natural" affinity between computer and management science. As a consequence, the procedural paradigm continues to dominate research efforts to understand the organization of office work.

Affiliated with the procedural paradigm are certain persistent troubles, however, which appear in management and computer science alike. These troubles are commonly attributed to the stubbornly ambiguous and imprecise properties of office procedures when compared, for example, to the step-wise instructions of computer science. As a consequence of those properties, implementation of a procedure in a given instance is a problematic enterprise. In this paper, I take the problematics of procedural implementation as an irremediable fact, and argue that rather than attempting to do away with the uncertainties of procedural specifications (an endless task), we understand those uncertainties as a consequence of the relationship of any principled instructions to the actual occasions of their use. The topic for research, then, is how it is that practical reasoning about procedures, finding their "definite meaning" is a constituent feature of the work of getting them done. By making that work a topic for office research, rather than the procedural specifications per se, certain long-standing troubles of office modeling may begin to yield.

Programmatic grounds for understanding the problematic relation of procedural instructions to their implementation are found in the sociological literature, specifically in Garfinkel and Sacks' "Formal Structures of Practical Actions" (1970). In particular, the problematics are located in the "indexical properties" of natural language descriptions in general. These properties mean that the "definite meaning" of the description (or in the case of the office, procedural instruction) is found, on each occasion of its use, by inspecting the circumstances which it is understood to describe, and only in that way. In principle, office work as practical action is this project of finding the "definite meaning" of a procedural instruction (where definite meaning is understood as the actual work involved in carrying it out) with respect to a particular case at hand.

Empirically, the problematics of procedural implementation and their solution are demonstrated in an example drawn from observation of an accounts payable auditing procedure. The data are examined for how it is that the evidence provided by documents, co-workers and clients is used, in conjunction with knowledge of the accounts payable procedure, to generate a record of action "according to procedure." A contrast case, wherein this work appears to be deferred, provides a further indication of the difference between procedural office work and the "execution" of procedural instructions.

This research is part of work in progress towards a Ph.D. in Social Anthropology from the University of California at Berkeley.

Reference:

Garfinkel, H. and H. Sacks  
 Formal Structures of Practical Actions  
 in *Theoretical Sociology*  
 J. McKinney and E. Tiryakian (eds.), Appleton-Century-Crofts, 1970.

Additional reference:

Suchman, L. and E. Wynn  
*Procedures and Problems in the Office Environment*  
 Xerox Internal Report, April 1979.

## 15. Additional Discussions

In addition to the discussions described by the abstracts above other discussions were led by some of the attendees. Below is a list of the attendees that led these discussions and a brief description of the topics covered.

- **Raymond E. Barber.** Dr. Barber presented an outline of the office information services in use in the Executive Office of the President. He described in house approaches to satisfying the criteria of (a) quick access to large amounts of data and (b) usability by computer naive operators.
- **Michael Hammer.** Professor Hammer discussed the Office Specification Language under development by his group at MIT. The goals and a field test of the language in several major corporations in the fall of 1980 were described. He briefly described the document preparation system (ETUDE) under development by his group and the philosophy behind its design.
- **Najah Naffah.** Dr. Naffah described the KAYAK project, a French national project for office automation at INRIA. The architecture of a workstation network was described. A system of layered protocols was presented with the aim of integrating services for the office.
- **Dionysis Tsichritzis.** Professor Tsichritzis described the electronic forms system developed at the University of Toronto. The development of the electronic forms system on top of a relational database system was described as well as some of the theory underlying the system.

## I. The Participants

The workshop participants and their respective organizations are listed below in alphabetical order.

Robert H. Anderson  
Interactive Systems Corporation  
1212 Seventh Street  
Santa Monica, CA 90401

Gerald R. Barber  
MIT Laboratory for Computer Science  
545 Technology Square - Room 812  
Cambridge, MA 02139

Raymond E. Barber  
Executive Office of the President  
Office of Administration  
Washington, DC 20503

Jack R. Buchanan  
Harvard Business School  
Boston, MA 02136

Peter de Jong  
IBM T. J. Watson Research Center  
Box 218  
Yorktown Heights, NY 10598

Clarence Ellis  
Xerox Palo Alto Research Center  
3333 Coyote Hill Road  
Palo Alto, CA 94304

Richard E. Fikes  
Xerox Palo Alto Research Center  
3333 Coyote Hill Road  
Palo Alto, CA 94304

V. Ellen Golden  
MIT Laboratory for Computer Science  
545 Technology Square - Room 831  
Cambridge, MA 02139

Ira P. Goldstein  
Xerox Palo Alto Research Center  
3333 Coyote Hill Road  
Palo Alto, CA 94304

Irene Grief  
MIT Laboratory for Computer Science  
545 Technology Square - Room 522  
Cambridge, MA 02139

Michael Hammer  
MIT Laboratory for Computer Science  
545 Technology Square - Room 217  
Cambridge, MA 02139

Carl Hewitt  
MIT Laboratory for Computer Science  
545 Technology Square - Room 813  
Cambridge, MA 02139

Robert Hudyma  
Computer Systems Research Group  
University of Toronto  
121 St. Joseph Street  
Toronto, Ontario, Canada M5S1A1

Reinhard Kofer  
SIEMENS AG  
ZFE-FL-SAR 112  
Otto Hahn Ring 6  
8 Muenchen 83  
WEST GERMANY

J. C. R. Licklider  
 MIT Laboratory for Computer Science  
 545 Technology Square - Room 218A  
 Cambridge, MA 02139

John O. Limb  
 Bell Laboratories  
 600 Mountain Avenue  
 Murray Hill, NJ 07974

Howard L. Morgan  
 Dept. of Decision Science  
 The Wharton School / CC  
 Philadelphia, PA 19104

Najah Naffah  
 INRIA - Project KAYAK  
 78150 Rocquencourt  
 FRANCE

William Newman  
 Logica VTS Ltd.  
 64 Newman Street  
 London W. 1.  
 England

Ian Page  
 Computer Systems Laboratory  
 Queen Mary College  
 Mile End Road  
 London, England E14NS

Sabine Rohlfis  
 Bell-Northern Research  
 P. O. Box 3511, Station C  
 Ottawa K1G4H7, Canada

Lucy A. Suchman  
 Xerox Palo Alto Research Center  
 3333 Coyote Hill Road  
 Palo Alto, CA 94304

Dionysis Tsichritzis  
 Computer Systems Research Group  
 University of Toronto  
 121 St. Joseph Street  
 Toronto, Canada M5S1A1

Michael Hammer  
 MIT Laboratory for Computer Science  
 545 Technology Square - Room 217  
 Cambridge, MA 02139

Robert Hudys  
 Computer Systems Research Group  
 University of Toronto  
 121 St. Joseph Street  
 Toronto, Ontario, Canada M5S1A1

Reinhard Koler  
 SIEMENS AG  
 ZEE PL. SAR 112  
 Otto Hahn Ring 8  
 8 München 83  
 WEST GERMANY